取扱説明書 - 日本語

FUJITSU

FUJITSU Software ServerView Suite

Server Management
Job API Tandem 1.5

2010 年 2 月版

製品名称の表記

本書では、本文中の製品名称を、次のように略して表記します。

製品名称	本文中の表記	
Microsoft® Windows Server® 2008 Standard Microsoft® Windows Server® 2008 Enterprise Microsoft® Windows Server® 2008 Datacenter Microsoft® Windows Server® 2008 Foundation Microsoft® Windows® Small Business Server 2008 Standard Microsoft® Windows® Small Business Server 2008 Premium	Windows Server 2008	Windows
Microsoft® Windows Server® 2008 R2 Standard Microsoft® Windows Server® 2008 R2 Enterprise Microsoft® Windows Server® 2008 R2 Datacenter Microsoft® Windows Server® 2008 R2 Foundation Microsoft® Windows® Web Server 2008 R2	Windows Server 2008 R2	
Microsoft® Windows Server® 2003, Standard Edition Microsoft® Windows Server® 2003, Enterprise Edition Microsoft® Windows Server® 2003, Enterprise Edition for Itanium-based Systems Microsoft® Windows® Small Business Server 2003	Windows Server 2003	
Microsoft® Windows Server® 2003, Standard x64 Edition Microsoft® Windows Server® 2003, Enterprise x64 Edition	Windows Server 2003 x64	
Microsoft® Windows Server® 2003 R2 Standard Edition Microsoft® Windows Server® 2003 R2 Enterprise Edition Microsoft® Windows® Small Business Server 2003 R2 Microsoft® Windows® Storage Server 2003 R2, Standard Edition	Windows Server 2003 R2	
Microsoft® Windows Server® 2003 R2 Standard x64 Edition Microsoft® Windows Server® 2003 R2 Enterprise x64 Edition	Windows Server 2003 R2 x64 または Windows Server 2003 R2	

製品名称	本文中の表記	
Red Hat Enterprise Linux 5	Red Hat Linux	Linux
	RHEL5	
Red Hat Enterprise Linux AS（v.4）	RHEL4	
Red Hat Enterprise Linux ES（v.4）		
SUSE Linux Enterprise Server 11	SuSE Linux	
	SuSE Linux SLES11 または SLES11	
SUSE Linux Enterprise Server 10	SuSE Linux SLES10 または SLES10	
VMware ESX 4	ESX4	VMware
VMware ESX 3.5	ESX3.5	

著作権および商標

Copyright © 2010 Fujitsu Technology Solutions GmbH.

All rights reserved

Microsoft、Windows、Windows Server、Hyper-V は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です。

Red Hat および Red Hat をベースとしたすべての商標とロゴは、米国およびその他の国における Red Hat, Inc. の商標または登録商標です。

BrightStor, ARCserve は、CA, Inc の登録商標です。

VMware、VMware ロゴ、VMware ESXi、VMware SMP および VMotion は VMware,Inc の米国およびその他の国における登録商標または商標です。

その他の各製品名は、各社の商標、または登録商標です。

その他の各製品は、各社の著作物です。

目次

1 はじめに 9

1.1 本書の適用範囲 9

1.2 本書の表記について 10

2 アーキテクチャ 11

3 セキュリティ 13

4 API クライアント／サーバ関数フロー 15

4.1 インストール、イメージ作成の使用例 15

5 API 関数クラス 21

5.1 関数の概要 21

5.2 データの種類 26

5.3 API 管理 28

5.3.1 GetAPIVersion 28

5.3.2 GetAPIinfo 29

5.3.3 CreateUserKey 32

5.4 サーバ／ノードリスト管理 33

5.4.1 AddServer 33

5.4.2 ChangeServerConfiguration 40

5.4.3 StartSystemInfoRetrieval 45

5.4.4 GetSystemInfoResult 48

5.4.5 RdDeplServerList 49

5.4.6 ImportDeploymentConfiguration 55

5.4.7 ExportDeploymentConfiguration 57

5.4.8 SetUserKey 59

5.4.9 RemoveUserKey 60

5.5 電源管理 61
5.5.1 SetPower 61
5.5.2 SetPXEBootImage 65
5.5.3 GetPower 68
5.5.4 GetBootMode 70

5.6 リポジトリ管理 72
5.6.1 AddRepository 74
5.6.2 RemoveRepository 76
5.6.3 GetRepositoryList 77
5.6.4 CreateImage 79
5.6.5 DeleteImage 87
5.6.6 GetRepositoryContents 88
5.6.7 ReadImagefilenameList 90
5.6.8 ReadImageInfoText 91
5.6.9 GetImageProperties 92

5.7 ディレクトリ管理 93
5.7.1 CreateDirectory 93
5.7.2 DeleteDirectory 95
5.7.3 GetDirectoryContents 97

5.8 グループ管理 99
5.8.1 CreateGroup 100
5.8.2 DeleteGroup 102
5.8.3 GetGroupList 103
5.8.4 SetGroupProperties 105
5.8.5 GetGroupProperties 107
5.8.6 AddGroupMember 108
5.8.7 RemoveGroupMember 110
5.8.8 GetGroupMemberList 111
5.8.9 SetGroupMemberProperties 113
5.8.10 GetGroupMemberProperties 115

5.9 デプロイメント管理 117
5.9.1 Deploy 118

5.10 ジョブ管理 130
5.10.1 GetJobStatusDetails 131
5.10.2 GetServerJobStatus 134
5.10.3 CancelJob 137
5.10.4 GetScheduledJobProperties 139
5.10.5 GetScheduledJobLogList 141
5.10.6 DeleteScheduledJob 143

6 ジョブスケジューリング 145

7 API エラーコード 149

8 インストール 151

8.1 システム要件 151

8.2 インストール 152

8.3 JobAPI と Internet Information Server (IIS) 158

8.3.1 Windows Server 2003 での JobAPI と Internet Information Server (IIS) 158

8.3.2 Windows Server 2008 での JobAPI と Internet Information Server (IIS) 158

8.4 インストールテスト 159

9 その他の情報 161

9.1 URL 161

9.2 HTTP 認証 162

9.3 HTTPS / SSL 164

9.4 SOAP タイムアウト 166

9.5 CGI タイムアウト 167

9.5.1 Apache 構成 167

9.5.2 IIS 構成 169

9.6 SOAP エラーの対処方法 171

9.7 SOAP デバック 172

9.7.1 Job API クライアント 172

9.7.2 Job API サーバ 172

10 Job API クライアントの開発 173

10.1 C/C++ スタブの生成 174

10.2 生成された C/C++ スタブの使用 177

索引 179

1 はじめに

Tandem は、ServerView Suite の強力なリモート インタフェースです。
Tandem に含まれている主要な API パッケージは次の 2 つです。

– central NodeList API （まだリリースされていません）

– central Job API

Job API および NodeList API は、SOAP に基づいた Web サービスとして実行されます。Simple Object Access Protocol（SOAP）は XML ベースのメッセージプロトコルです。RPC 式の要求応答ダイアログを実行するために使用する構成メッセージに対し、一連の規則を定義します。SOAP メッセージは OS、プログラム言語、またはトランスポートプロトコルに依存しません。したがって、要求応答ダイアログ内のクライアントとサーバは、SOAP メッセージの準備と処理ができる限り、どんなプラットフォーム上でも動作し、どんな言語でも書き込むことができます。

Job API Tandem は、Deployment Manager フロントエンドが提供する関数のほとんどを提供するように設計されています。Job API を使って単一の、またはグループ化されたサーバの準備、インストール、およびクローニングができます。Job API は ServerView Deployment Manager サービスに対してジョブを提示します。

Job API 関数クラスのサーバ／ノードリストの管理には関数が含まれています。その関数を使って、ServerView Suite ソフトウェアで管理されている管理サーバすべての物理ビューを含んだ集中 ServerView サーバリストを操作します。そのため、これまで ServerView により管理されていなかったベアサーバを追加できます。Nodelist API がリリースされるまでは、サーバ／ノードリストの管理が代用の関数を提供します。

Job API は Deployment Manager CD に収録されている独立したインストールパッケージです。

1.1 本書の適用範囲

本書は Job API Tandem のパラメータと様々な関数で予測される動作について説明します。本書は、LAN 上からリモートで、または第三者のソフトウェアを使って富士通サーバ管理関数を使用したいプログラマーを対象としています。

1.2 本書の表記について

本書の表記方法は次のとおりです。

IP アドレスは、標準的な形式「XXX.XXX.XXX.XXX」を使って表記します。すなわち、32 ビット整数は、点で区切られた 0 から 255 までの 10 進数を 4 つ使用して指定します（RFC791）。

有効な GUID 形式は「XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX」です。X には 16 進法の数字（0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F）が入ります。

関数内の MAC アドレスは形式「AA-BB-CC-DD-EE-FF」を使って返されます。入力パラメータとして MAC アドレスが必要な場合はいつでもこの「AA-BB-CC-DD-EE-FF」形式を使って追加できます。大文字、小文字を区別する必要はありません。

2 アーキテクチャ

次の図は、Tandem の Job API および NodeList API パッケージと組み合わせた ServerView Suite のサーバ管理に関する概要を示します。

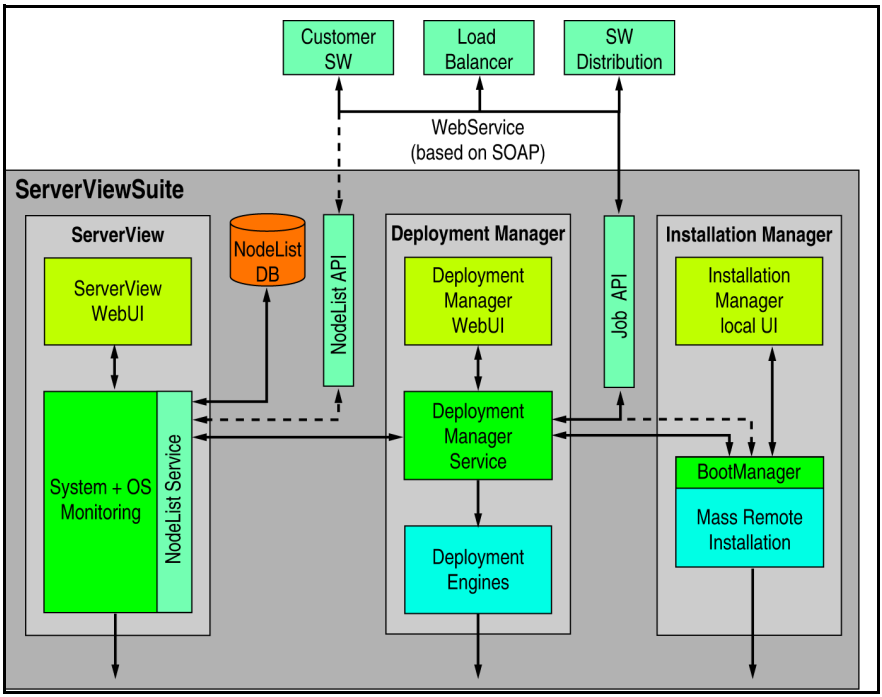
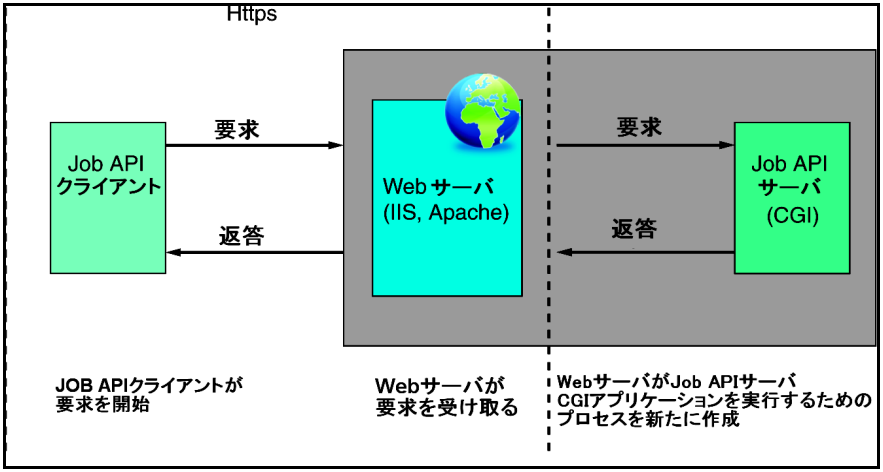


図 1: アーキテクチャ

3 セキュリティ

現行の Job API サービスは CGI ベースの Web アプリケーションとして実現されます。原則として、Web サーバの視点からみた CGI の動作とは、特定の Web サービスが定義され、CGI アプリケーションにより提供されるものです。Web サーバへの要求を受け取ると、クライアントが入力値として送信したデータとともに、CGI アプリケーションが呼び出されます。最終的には、CGI アプリケーションからの出力は Web サーバによって収集され、クライアントへ返信されます。

この機能により、SOAP メッセージに含まれている情報はクライアントと Web サーバ間のみで傍受されます。最下層では、SOAP メッセージが HTTPS へ渡されます。HTTPS は通信に SSL を使用するため、スヌーピングを防ぐ目的でメッセージ内容は暗号化され、クライアントとサーバはお互いの身元を確認します。したがって、Web サーバと Job API クライアント（CGI ベースの Web アプリケーションそのものではありません）に対して HTTPS/SSL サポートを設定する必要があります。



4 API クライアント／サーバ関数フロー

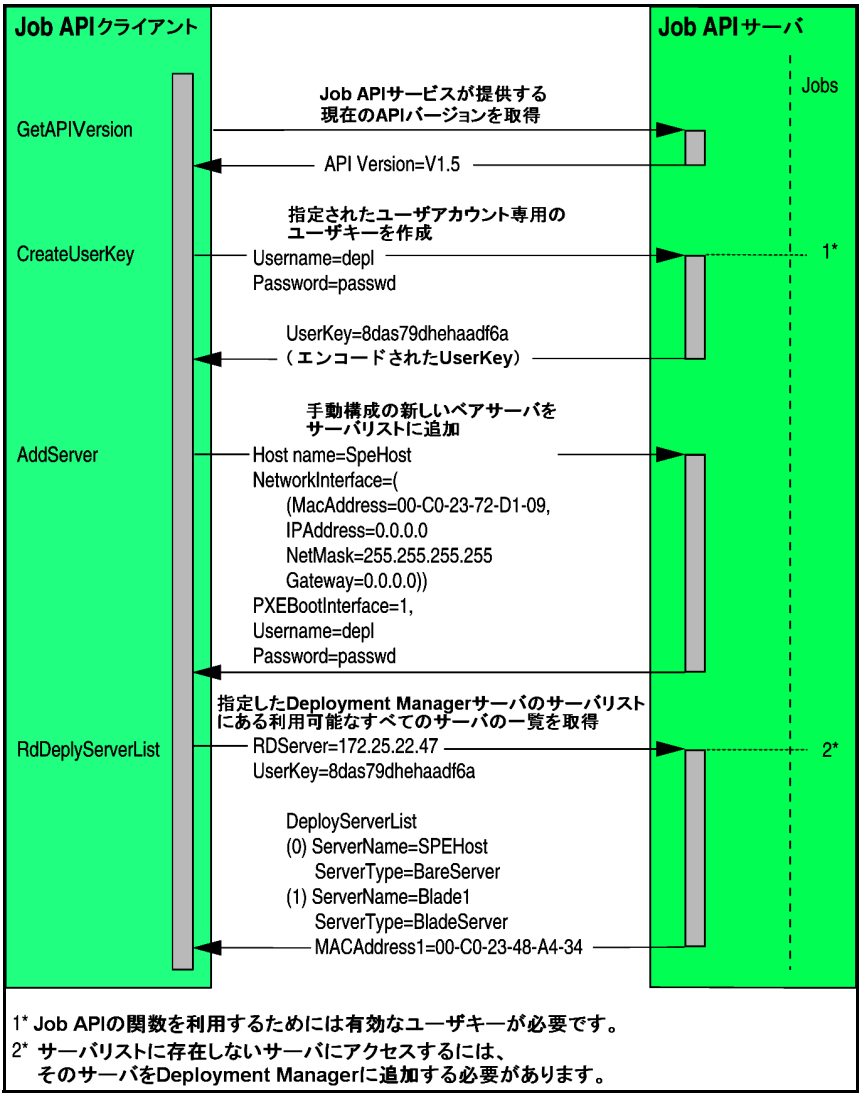
一般的にコンピュータサイエンスでは、プロトコルとは、フォーマット、内容、目的、および 2 つのエンドポイント間の通信シーケンスを特定する一連の規則であると説明されています。したがって、Job API クライアントと Job API サーバ間の情報交換をプロトコルとして解釈できますが、クライアント／サーバ間の通信は Job API 仕様内で関数フローとして定義され、関数を呼び出す順序は設定されません。ただし、最初に Job API クライアントに対して有効なユーザキーを作成する必要があります。ユーザキーは何度でも作成できますが、毎回新しいものを作成する必要はなく、Job API クライアントに対しては 1 回のみでかまいません。

次の節で、Job API 関数の用途と正しいシーケンスについて説明します。

4.1 インストール、イメージ作成の使用例

最初の使用例は、対象システムのインストールおよび他のサーバへのデプロイメントに使用するイメージの作成に関して説明します。

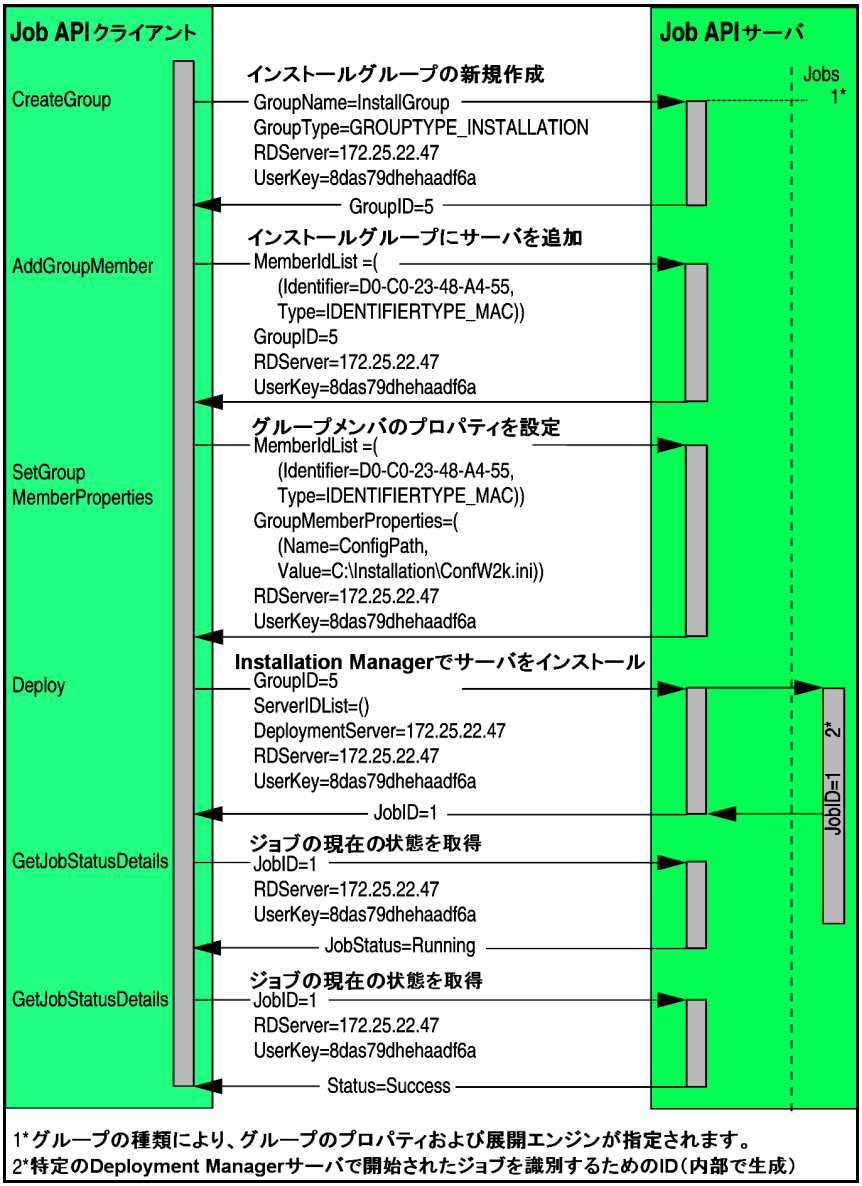
前述のとおり、Job API 関数すべてを使用するには、有効なユーザキーを作成する必要があります。すべての関数を使用するためには、UserKey が必要で、このパラメータで有効なユーザキーを指定する必要もあります。通常、サーバリストにあるすべてのサーバは、種類にかかわらず Job API 上で管理できます。しかし、ServerView により検出されたサーバ、および *AddServer* により新しく追加されたベアサーバの一覧を取得するには、関数 *RdDeplServerList* を使用します。

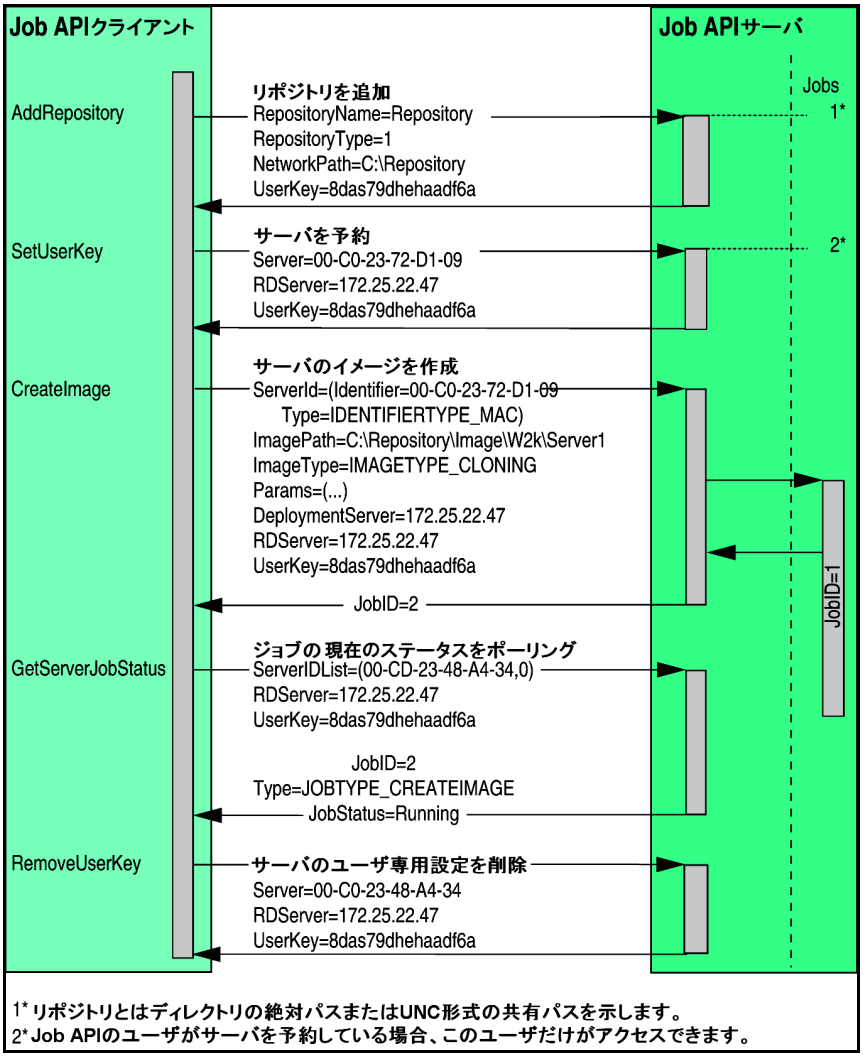

Job APIクライアント
Job APIサーバ
Jobs
GetAPIVersion
Job APIサービスが提供する
現在のAPIバージョンを取得
API Version=V1.5
CreateUserKey
指定されたユーザアカウント専用の
ユーザキーを作成
Username=depl
Password=passwd
1*
UserKey=8das79dhehaadf6a
（エンコードされたUserKey）
AddServer
手動構成の新しいベアサーバを
サーバリストに追加
Host name=SpeHost
NetworkInterface=(
(MacAddress=00-C0-23-72-D1-09,
IPAddress=0.0.0.0
NetMask=255.255.255.255
Gateway=0.0.0.0))
PXEBootInterface=1,
Username=depl
Password=passwd
RdDeplyServerList
指定したDeployment Managerサーバのサーバリスト
にある利用可能なすべてのサーバの一覧を取得
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
2*
DeployServerList
(0) ServerName=SPEHost
ServerType=BareServer
(1) ServerName=Blade1
ServerType=BladeServer
MACAddress1=00-C0-23-48-A4-34
1* Job APIの関数を利用するためには有効なユーザキーが必要です。
2* サーバリストに存在しないサーバにアクセスするには、
そのサーバをDeployment Managerに追加する必要があります。

例えば、対象システムとして新しいベアサーバをインストールするには、コンフィグレーションファイルのパスをパラメータとして直接指定する、またはインストールグループを作成します。インストールグループを作成した場合、グループメンバ固有のプロパティは、あらかじめ Installation Manager で定義されたコンフィグレーションファイルのパスになります。

Installation Manager は deploy の呼び出しによって指定したグループ内、またはサーバのサブグループ内のサーバ（パラメータ *ServerIdList* をご覧ください）のどちらかをインストールします。インストール作業が完了した後（*GetJobStatusDetails* を使ってジョブの状態をポーリングします）、他のサーバへデプロイメントするために、インストールされた対象システムのイメージが作成されます。Job API と Deployment Manager はリポジトリ内にある利用可能なすべてのイメージを管理します。つまり、新しいクローニングイメージを可視化するには、*AddRepository* を使って *REPOSITORYTYPE_CLONING* タイプの新しいリポジトリをリポジトリ管理へ追加します。

基本的に、各サーバは様々な Job API クライアントや Deployment Manager フロントエンドによって管理されるため、*SetUserKey* を使用してサーバ専用のユーザキーを設定しておくことを推奨します。インストールとイメージ作成作業が正常に完了した後、この対象システムの専用設定を必ず削除してください。削除は関数 *RemoveUserKey* を使って実行します。


Job APIクライアント
Job APIサーバ
Jobs
1*
CreateGroup
インストールグループの新規作成
GroupName=InstallGroup
GroupType=GROUPTYPE_INSTALLATION
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
GroupID=5
AddGroupMember
インストールグループにサーバを追加
MemberIdList =(
(Identifier=D0-C0-23-48-A4-55,
Type=IDENTIFIERTYPE_MAC))
GroupID=5
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
SetGroup
MemberProperties
グループメンバのプロパティを設定
MemberIdList =(
(Identifier=D0-C0-23-48-A4-55,
Type=IDENTIFIERTYPE_MAC))
GroupMemberProperties=(
(Name=ConfigPath,
Value=C:\Installation\ConfW2k.ini))
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
Deploy
Installation Managerでサーバをインストール
GroupID=5
ServerIDList=()
DeploymentServer=172.25.22.47
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
JobID=1
2*
JobID=1
GetJobStatusDetails
ジョブの現在の状態を取得
JobID=1
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
JobStatus=Running
GetJobStatusDetails
ジョブの現在の状態を取得
JobID=1
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
Status=Success
1*グループの種類により、グループのプロパティおよび展開エンジンが指定されます。
2*特定のDeployment Managerサーバで開始されたジョブを識別するためのID（内部で生成）


Job APIクライアント
Job APIサーバ
Jobs
1*
2*
AddRepository
リポジトリを追加
RepositoryName=Repository
RepositoryType=1
NetworkPath=C:\Repository
UserKey=8das79dhehaadf6a
SetUserKey
サーバを予約
Server=00-C0-23-72-D1-09
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
CreateImage
サーバのイメージを作成
ServerId=(Identifier=00-C0-23-72-D1-09
Type=IDENTIFIERTYPE_MAC)
ImagePath=C:\Repository\Image\W2k\Server1
ImageType=IMAGETYPE_CLONING
Params=(...)
DeploymentServer=172.25.22.47
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
JobID=2
JobID=1
GetServerJobStatus
ジョブの 現在のステータスをポーリング
ServerIDList=(00-CD-23-48-A4-34,0)
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
JobID=2
Type=JOBTYPE_CREATEIMAGE
JobStatus=Running
RemoveUserKey
サーバのユーザ専用設定を削除
Server=00-C0-23-48-A4-34
RDServer=172.25.22.47
UserKey=8das79dhehaadf6a
1* リポジトリとはディレクトリの絶対パスまたはUNC形式の共有パスを示します。
2* Job APIのユーザがサーバを予約している場合、このユーザだけがアクセスできます。

5 API 関数クラス

次の章では、Job API の様々な関数のパラメータと予測される動作に関する非公式な説明を記載します。すべてのパラメータは SOAP/XML で暗号化されます。

5.1 関数の概要

次の表では、実装されている関数の概要を示します。

API 管理

GetAPIVersion	Job API のバージョンに関する情報を返します。28 ページをご覧ください。
GetAPIInfo	Job API と使用可能なデプロイメントエンジンに関する詳細情報を返します。29 ページをご覧ください。
CreateUserKey	特有のユーザキーを作成します。32 ページをご覧ください。

サーバ／ノードリスト管理

AddServer	指定したデプロイメント構成を使って新しいサーバを追加します。33 ページをご覧ください。
ChangeServerConfiguration	指定したサーバのデプロイメント構成設定を変更します。40 ページをご覧ください。
StartSystemInfoRetrieval	指定したサーバからシステム情報の詳細を取得するジョブを起動します。45 ページをご覧ください。
GetSystemInfoResult	指定したサーバからシステム情報の詳細を返します。48 ページをご覧ください。
RdDeplServerList	デプロイメントテーブルにある使用可能なサーバすべての一覧を返します。49 ページをご覧ください。
ImportDeploymentConfiguration	デプロイメントテーブルを XML 文書形式でインポートします。55 ページをご覧ください。
ExportDeploymentConfiguration	デプロイメントテーブルを XML 文書形式でエクスポートします。57 ページをご覧ください。

SetUserKey	所定のユーザキーに対して 1 つ以上のサーバを予約します。59 ページをご覧ください。
RemoveUserKey	指定したサーバのユーザ予約を削除します。60 ページをご覧ください。

電源管理

SetPower	サーバの電源を投入、または切断します。61 ページをご覧ください。
SetPXEBootImage	TFTP サーバ上の所定ブートイメージを設定し、そのサーバで PXE ブートが行えるようにします。65 ページをご覧ください。
GetPower	サーバの電源スイッチに対する現在の設定を判断します。68 ページをご覧ください。
GetBootMode	サーバの現在のブートモードを取得します。70 ページをご覧ください。

リポジトリ管理

AddRepository	新しいリポジトリを追加します。74 ページをご覧ください。
RemoveRepository	指定したリポジトリを削除します。76 ページをご覧ください。
GetRepositoryList	すべてのリポジトリまたは、任意タイプのリポジトリに関する詳細情報を一覧で返します。77 ページをご覧ください。
CreateImage	複数の対象システムのデプロイメントに対して使用可能なイメージを作成します。79 ページをご覧ください。
DeleteImage	既存のイメージファイルを削除します。87 ページをご覧ください。
GetRepositoryContents	指定したリポジトリ、または指定した種類のすべてのリポジトリ内で検出されたイメージファイル名の一覧を返します。88 ページをご覧ください。
ReadImagefilenameList	既知リポジトリすべてで検出されたイメージファイル名の一覧を返します。90 ページをご覧ください。
ReadImageInfoText	指定したイメージの info テキストを返します。91 ページをご覧ください。
GetImageProperties	所定イメージのプロパティを XML 文書形式で返します。92 ページをご覧ください。

ディレクトリ管理

CreateDirectory	新しいディレクトリを作成します。93 ページをご覧ください。
DeleteDirectory	ディレクトリを削除します。95 ページをご覧ください。
GetDirectoryContents	指定したディレクトリに含まれている既知ファイルすべてを一覧で返します。97 ページをご覧ください。

グループ管理

CreateGroup	指定した種類の新しいグループを作成します。100 ページをご覧ください。
DeleteGroup	指定したグループを削除します。102 ページをご覧ください。
GetGroupList	既知グループすべてを一覧で返します。103 ページをご覧ください。
SetGroupProperties	所定のグループに対して、指定したグループのプロパティ値を設定します。105 ページをご覧ください。
GetGroupProperties	使用可能なグループのプロパティとその値に関する情報を返します。107 ページをご覧ください。
AddGroupMember	所定グループに新しいサーバを追加します。108 ページをご覧ください。
RemoveGroupMember	所定グループから、指定したサーバを削除します。110 ページをご覧ください。
GetGroupMemberList	指定したグループに所属する既知メンバを一覧で返します。111 ページをご覧ください。
SetGroupMemberProperties	指定したグループメンバのプロパティを指定値へ設定します。113 ページをご覧ください。
GetGroupMemberProperties	使用可能なサーバプロパティとその値に関する情報を返します。115 ページをご覧ください。

デプロイメント管理

Deploy	指定した対象サーバをデプロイメントし、またこれらサーバのブートを開始します。118 ページをご覧ください。

ジョブ管理

GetJobStatusDetails	所定の JobID を使って現在のジョブ状態と詳細を返します。131 ページをご覧ください。
GetServerJobStatus	所定のサーバ識別子リストを使って現在のジョブ状態を返します。134 ページをご覧ください。
CancelJob	実行中のジョブをキャンセルします。137 ページをご覧ください。
DeleteScheduledJob	所定の ScheduledJobID を使って既存のスケジュールジョブを削除します。143 ページをご覧ください。
GetScheduledJobProperties	所定のスケジュールジョブのプロパティすべてを返します。139 ページをご覧ください。
GetScheduledJobIDList	所定のスケジュールジョブの中で実行中または実行済みであるジョブの一覧を返します。141 ページをご覧ください。

5.2 データの種類

本書で使用している一般的なデータの種類は次のとおりです。

種類	説明
string	アルファベットや数値を含む文字列です。
int	整数です。
long	長整数です。

表 1: データの種類

ArrayOf_PropertyData 構造体のレイアウト

プロパティの構造体のレイアウトは次のとおりです。

Name
: プロパティの既知名を指定します。

Value
: 指定したプロパティの値を設定します。

ServerIdentifier 構造体のレイアウト

サーバを判別するには、サーバの MAC アドレス、IP アドレスまたは、ホスト名を指定します。次の構造体を使ってサーバの種類から、サーバを判別します。

```
Identifier=<string> Type=<int>
```

Identifier
: 既知サーバ特有の識別子を含む文字列です。

Type
: 識別子の種類です。
0 (IDENTIFIERTYPE_MAC): ネットワークインタフェースの物理アドレスです。
1 (IDENTIFIERTYPE_IP): サーバの IP アドレスです。これは、*RdDeplServerList* により返された *SV_IPAddress* である必要があります。
2 (IDENTIFIERTYPE_NAME): サーバのホスト名です。これは、*RdDeplServerList* により返された *ServerName* である必要があります。

ArrayOf_ParameterData 構造体のレイアウト

次の構造体はパラメータを指定します。

```
Name=<string> Value=<string>
```

Name
: パラメータの名前です。

Value
: 該当するパラメータ名の値です。

5.3 API 管理

Job API 関数クラスの「API 管理」は、Job API の使用を管理するため、つまり Job API Web サービスや使用されているデプロイメントエンジンに関する情報を取得するために使用します。

5.3.1 GetAPIVersion

関数 *GetAPIVersion* を使って、Job API サービスが提供する最新の API バージョンを API の呼び出し元へ通知します。

構文

```
GetAPIVersion UserKey=<string>
```

パラメータ

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

APIVersion
: Job API の現在のバージョンです（例：1.5）。

5.3.2 GetAPIinfo

GetAPIVersion が返すバージョン情報とは別に、*GetAPIInfo* は Job API やデプロイメント環境についての詳細（例：ビルド番号、使用可能なデプロイメントエンジンなど）を取得します。

この関数を使用するには、Deployment Manager が必要です。

構文

```
GetAPIInfo
    DeploymentEngines= <ArrayOf_DeploymentEngineIdentifier>
    RDServer=<string>
    UserKey=<string>
    APIProperties=<ArrayOf_PropertyData>
    DeploymentEngineProperties=<ArrayOf_DeploymentEngineData>
```

パラメータ

Deployment Engines
: 現在のエンジンプロパティを返すために確認するデプロイメントエンジンを指定します。構造体に関する詳細は、30 ページの " デプロイメントエンジン識別子の構造体レイアウト " をご覧ください。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

APIProperties
: Job API プロパティすべての配列（例：*BuildNumber*）を返します。構造体に関する詳細は、26 ページの "ArrayOf_PropertyData 構造体のレイアウト " をご覧ください。

DeploymentEngineProperties
: 指定されたデプロイメントエンジン（例：*Installation Manager*）の状態を返します。構造体に関する詳細は、31 ページの " デプロイメントエンジンデータの構造体レイアウト " をご覧ください。

デプロイメントエンジン識別子の構造体レイアウト

デプロイメントエンジン識別子の構造体には、デプロイメントエンジンに関する詳しい接続情報が含まれています。この構造体のレイアウトは次のとおりです。

```
ServerId=<ServerIdentifier> Type=<int> Username=<string>
Password=<string>
```

ServerId
: エンジンがインストールされているサーバの名前です。構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

Type
: デプロイメントエンジンの種類です。
1: (DEPLOYMENTENGINE_SCW)
2: (DEPLOYMENTENGINE_INSTALLATIONMANAGER)
3: (DEPLOYMENTENGINE_DEPLOYMENTMANAGER)

Username
: 指定したエンジンのユーザアカウント名です。このパラメータは必須です。

Password
: 指定したユーザアカウントに対するパスワードです。このパラメータは必須です。

デプロイメントエンジンデータの構造体レイアウト

デプロイメントエンジンデータの構造体には、指定した複数のエンジンに関するリターン情報が含まれています。この構造体のレイアウトは次のとおりです。

```
ServerId=<ServerIdentifier> Type=<int> Properties=
<ArrayOfPropertyData>
```

ServerId

エンジンがインストールされているサーバを表示します。構造体に関する詳細は 26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

Type

デプロイメントエンジンの種類を表示します。
1: (DEPLOYMENTENGINE_SCW)
2: (DEPLOYMENTENGINE_INSTALLATIONMANAGER)
3: (DEPLOYMENTENGINE_DEPLOYMENTMANAGER)

Properties

指定したデプロイメントエンジンのプロパティを表示します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.3.3 CreateUserKey

関数 *CreateUserKey* 指定したユーザアカウント専用のユーザキーを作成します。Job API を使って作業するには、この有効なユーザキーが必要です。

構文

```
CreateUserKey Username=<string> Password=<string>
```

パラメータ

Username

Deployment Admins グループに含まれるユーザアカウント名を指定します。

Password

指定したユーザアカウントに対するパスワードを指定します。

パラメータ *username* および *password* の長さと形式は OS によって異なります。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

UserKey

指定したユーザアカウント専用のユーザキーです。

5.4 サーバ／ノードリスト管理

サーバ／ノードリスト管理は ServerView により検出されたサーバや、Deployment Manager または Job API により追加されたサーバに関する構成、状態、またその他の情報を提供します。これらサーバは、サーバの種類にかかわらず Job API によって管理されます。

5.4.1 AddServer

Job API の関数 *AddServer* を使って、新しいベアサーバに対する構成設定（指定したデプロイメント構成を使って）を手動で入力します。

構文

```
AddServer Hostname=<string> GUID=<string>
          NetworkInterface=<ArrayOf_NetworkInterfaceData>
          PXEBootInterface=<int>
          Params=<ArrayOf_ParameterData>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

Hostname

指定したサーバの新しい名前を指定します。このサーバをインストールする、またはイメージでデプロイメントする場合、この新しい名前は新規サーバのホスト名として使用されます。
このパラメータは必須です。

GUID

グローバル一意識別子を指定します。
デフォルト値：00000000-0000-0000-000000000000

NetworkInterface

新しいサーバのネットワークインタフェースに関する詳細が含まれている *NetworkInterfaceData* タイプの構造体の配列を指定します。ネットワーク構成に関する詳細は、35 ページの "NetworkInterfaceData 構造体のレイアウト " をご覧ください。

PXEBootInterface

PXE ブート（クローニングおよびインストール用）に使用するネットワークインタフェースのインタフェースインデックス（1 で始まる）を指定します。PXE ブートを使用するネットワークインタフェースを 1 つ指定するためには、このパラメータを必ず指定してください。

Params

ParameterData タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は、37 ページの " 使用可能なパラメータの表 " をご覧ください。また、パラメータの構造体に関しては、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

パラメータ *Hostname*、*DNSSuffix*、*AdminUsername* および *AdminPassword* の長さおよび許可されている文字は OS によって異なります。

NetworkInterfaceData 構造体のレイアウト

ネットワークインタフェースデータの構造体には、新しいサーバのネットワークインタフェース構成に関する詳細が含まれています。この構造体のレイアウトは次のとおりです。

```
Type=<int> Properties=<ArrayOf_PropertyData>
```

Type

ネットワークインタフェースの種類を指定します。
1: (NETWORKINTERFACETYPE_PHYSICAL)

Properties

新しいサーバのネットワークインタフェースに対するプロパティを指定します。26 ページの "ArrayOf_PropertyData 構造体のレイアウト " をご覧ください。

使用可能な物理ネットワークインタフェースプロパティの表

この表はネットワークインタフェースデータの配列で指定できるプロパティをすべて表示します。

MACAddress

新しいサーバのネットワークインタフェースの物理アドレスを指定します。このパラメータは必須です。

IPAddress

ネットワークインタフェースの IP アドレスを指定します。
DHCP を使用する場合、IP アドレスは 0.0.0.0. になります。
デフォルト値：0.0.0.0.

NetMask

ネットワークインタフェースのネットマスクを指定します。
DHCP を使用する場合、ネットマスクは 255.255.255.255 になります。
デフォルト値：255.255.255.255

Gateway

LAN のデフォルトゲートウェイを指定します。
DHCP を使用する場合、デフォルトゲートウェイの IP アドレスは 0.0.0.0. になります。
デフォルト値：0.0.0.0.

DNSServerList

コンマで区切られた DNS サーバ名またはドメインサーバの IP アドレス一覧を指定します。
デフォルト値："

RegisterInDNS

DNS で IP アドレスの動的登録を有効、または無効にします。このパラメータは Windows システムに対してのみ評価されます。
0: (OFF)
1: (ON)
デフォルト値：1 (ON)

UseSuffixInDNS

この接続の IP アドレスと接続固有のドメイン名を登録するための DNS 動的更新を有効、または無効にします。*RegisterInDNS* が有効の場合、ここで有効にした登録はコンピュータのフルネームに対する DNS 登録の追加項目となります。このパラメータは Windows システムに対してのみ評価されます。
0: (OFF)
1: (ON)
デフォルト値：0 (OFF)

ConnectionDNSSuffix

このインタフェースのドメインサフィックス文字列を指定します。このパラメータは Windows システムに対してのみ評価されます。
デフォルト値："

使用可能なパラメータの表

この表は、パラメータの配列で指定できるパラメータを表示します。

SystemType

サーバシステムの種類を指定します。例：PRIMERGY TX300
デフォルト値："Any Server"

ServerModel

富士通サーバモデルに関する標準名称を指定します。例えば D1710 とは詳細名称の S26361-D171-A1 から取ったものです。
デフォルト値：""

DNSSuffix

ドメインサフィックス文字列を指定します。例：<*hostname*>、<*dnssuffix*>
デフォルト値：""

DNSAppendList

プライマリおよび接続固有な DNS サフィックスの代わりにサフィックスを追記することで不適切な名前を解決するために設定した DNS サフィックスのリストを指定します。
デフォルト値：""

Community

SNMP サービス用に定義された SNMP コミュニティを指定します。これは、サーバリストで SNMP コミュニティ文字列を指定するためだけに使用し、このサーバの SNMP アクセスのために ServerView Suite アプリケーションにより使用されます。クローニングまたはインストールした SNMP サービスの SNMP コミュニティは定義しません。
デフォルト値：""

AdminUsername

管理者のユーザアカウントを指定します。
デフォルト値：""（Windows では *administrator*、Linux では *root* を使用します。）

AdminPassword

指定した管理者のユーザアカウントに対するパスワードを指定します。
デフォルト値：""（デプロイメント中はパスワードを変更しないでください。）

RemoteAccess

リモートアクセスを指定します。
0 (REMOTEACCESSTYPE-NONE)
1 (REMOTEACCESSTYPE-BMC): BMC サポート
2 (REMOTEACCESSTYPE-WOL): Wake on LAN
デフォルト値：0

IpmiMgmtIPAddress

BIOS セットアップで設定されているとおりにベースボード管理コントローラ（BMC）の IP アドレスを指定します。
デフォルト値："".
RemoteAccess が 1 (BMC) の場合、このパラメータは必須です。

IpmiUsername

BMC 認証用のユーザ名（Ipmi 1.5）を指定します。
デフォルト値："".
RemoteAccess が 1 (BMC) の場合、このパラメータは必須です。

IpmiPassword

BMC 認証用（Ipmi 1.5）ユーザアカウントに対するパスワードを指定します。
デフォルト値："".
RemoteAccess が 1 (BMC) の場合、このパラメータは必須です。

IpmiTimeout

BMC アクセスごとのタイムアウト値を秒単位で指定します。
デフォルト値：3

IpmiRetries

BMC アクセスごとの再試行回数を指定します。
デフォルト値：1

WOLBroadcastIPAddress

マジックパケットの送信先となる対象の IP アドレスを指定します。パラメータが空白で、*RemoteAccess* が *WOL* の場合、対象は同じ LAN セグメント内でなければなりません。リミテッドブロードキャスト IP アドレス 255.255.255.255 を使用し、マジックパケットはイーサネットブロードキャストとして UDP ポート 9 へ直接送信されます。
デフォルト値：""

WOLBootstrapProtocolIPAddress

ブートストラッププロトコルサーバの IP アドレスを指定します。
ブロードキャストが UDP ポート 67 へ送信され、*WOL BroadcastIPAddress* にユニキャスト IP アドレスが含まれている場合、このパラメータは必須です。
デフォルト値："

指定したサーバの種類が IPMI over LAN をサポートしない場合、Wake on LAN をデフォルトとして使用します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.4.2 ChangeServerConfiguration

この関数を使って、指定したサーバの現在の設定値（デプロイメント構成を含む）を変更します。定義したパラメータのみがサーバの構成を変更します。

構文

```
ChangeServerConfiguration ServerId=<ServerIdentifier>
          NetworkInterface=<ArrayOf_NetworkinterfaceData>
          Params=<ArrayOf_ParameterData>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerId

デプロイメント構成を変更するサーバのサーバ ID を指定します。構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。
このパラメータは必須です。

NetworkInterface

サーバのネットワークインタフェースに関する詳細を含む NetworkInterfaceData タイプの構造体の配列を指定します。ネットワークインタフェースの構造体に関する詳細は、35 ページの "NetworkInterfaceData 構造体のレイアウト " をご覧ください。物理ネットワークインタフェースについては、41 ページの " 使用可能な物理ネットワークインタフェースプロパティの表 " をご覧ください。

i *NetworkInterfaceData* を指定しない場合、サーバの現在のネットワークインタフェース構成は変更されません。しかし、*NetworkInterfaceData* を指定する場合は、すべてのネットワーク構成（すべてのインタフェース）を含めて指定してください。

Params

ParameterData タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は、43 ページの " 使用可能なパラメータの表 " をご覧ください。またパラメータ構造に関しては、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

現在のバージョンでは、指定したブレードに対してのみホスト 名を変更できます。今後のバージョンでは、サーバ全種類に対してホスト 名を変更できるようになります。

使用可能な物理ネットワークインタフェースプロパティの表

この表では、ネットワークインタフェースデータの配列内で指定できるプロパティを表示します。

MACAddress

変更するサーバのネットワークインタフェースの物理アドレスを指定します。

IPAddress

ネットワークインタフェースの IP アドレスを指定します。
DHCP を使用する場合、IP アドレスは 0.0.0.0 になります。
デフォルト値：0.0.0.0

NetMask

ネットワークインタフェースのネットマスクを指定します。
DHCP を使用する場合、ネットマスクは 255.255.255.255 になります。
デフォルト値：255.255.255.255

Gateway

LAN のデフォルトゲートウェイを指定します。
DHCP を使用する場合、デフォルトゲートウェイの IP アドレスは 0.0.0.0 になります。
デフォルト値：0.0.0.0

DNSServerList

コンマで区切られた DNS サーバ名またはドメインサーバの IP アドレスのリストを指定します。
デフォルト値："

RegisterInDNS

DNS で IP アドレスの動的登録を有効、または無効にします。このパラメータは Windows システムでのみ評価されます。
0: (OFF)
1: (ON)
デフォルト値：1 (ON)

UseSuffixInDNS

この接続の IP アドレスと接続固有のドメイン名を登録するために DNS 動的更新を有効または無効にします。*RegisterInDNS* が有効の場合、ここで有効にした登録はコンピュータのフルネームに対する DNS 登録の追加項目となります。
このパラメータは Windows システムに対してのみ評価されます。
0: (OFF)
1: (ON)
デフォルト値：0 (OFF)

ConnectionDNSSuffix

このインタフェースに対するドメインサフィックス文字列を指定します。
このパラメータは Windows システムに対してのみ評価されます。
デフォルト値："

使用可能なパラメータの表

この表には、パラメータの配列で指定できるパラメータを表示します。

パラメータの指定は任意です。したがって、定義したパラメータのみがサーバの構成を変更します。

Hostname
: 指定したサーバの新しい名前を指定します。このサーバをインストールする、またはイメージでデプロイメントする場合、この新しい名前は新規サーバのホスト名として使用されます。

GUID
: グローバル一意識別子を指定します。

SystemType
: サーバシステムの種類を指定します。例：PRIMERGY TX300

ServerModel
: 富士通サーバモデルに関する標準名称を指定します。例えば D1710 とは詳細名称の S26361-D171-A1 から取ったものです。

DNSSuffix
: ドメインサフィックス文字列を指定します。例：<*hostname*>、<*dnssuffix*>

Community
: SNMP サービス用に定義された SNMP コミュニティを指定します。これは、サーバリストの SNMP コミュニティ文字列を指定するためだけに使用し、このサーバの SNMP アクセスのために ServerView Suite アプリケーションにより使用されます。クローニングまたはインストール後、これは SNMP サービスの SNMP コミュニティを定義しません。

AdminUsername
: 管理者のユーザアカウントを指定します。

AdminPassword
: 指定した管理者のユーザアカウントに対するパスワードを指定します。

PXEBootInterface
: PXE ブート（クローニングおよびインストール用）に使用するネットワークインタフェースのインタフェースインデックス（1 で始まる）を指定します。PXE ブートに使用するネットワークインタフェースを必ず指定してください。

RemoteAccess
: リモートアクセスを指定します。
0 (REMOTEACCESSTYPE-NONE)
1 (REMOTEACCESSTYPE-BMC): BMC サポート
2 (REMOTEACCESSTYPE-WOL): Wake on LAN

IpmiMgmtIPAddress
: ベースボード管理コントローラ（BMC）の IP アドレスを指定します。

IpmiUsername
: BMC 認証（Ipmi 1.5）用のユーザ名を指定します。

IpmiPassword
: BMC 認証（Ipmi 1.5）用のユーザアカウントに対するパスワードを指定します。

IpmiTimeout
: BMC アクセスごとのタイムアウト値を秒単位で指定します。

IpmiRetries
: BMC アクセスごとの再試行回数を指定します。

WOLBroadcastIPAddress
: マジックパケットの送信先となる対象の IP アドレスを指定します。このパラメータが空白の場合、対象は同じ LAN セグメント内でなければなりません。リミテッドブロードキャスト IP アドレスの 255.255.255.255 を使って、マジックパケットをイーサネットブロードキャストとして UDP ポート 9 へ送信します。

WOLBootstrapProtocolIPAddress
: ブートストラッププロトコルサーバの IP アドレスを指定します。ブロードキャストが UDP ポート 67 へ送信され、また WOLBroadcastIPAddress にユニキャスト IP アドレスが含まれている場合、このパラメータは必須です。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.4.3 StartSystemInfoRetrieval

関数 *StartSystemInfoRetrieval* を使って、指定したサーバに関する現在のシステム情報を取得できます。様々な要因により、このジョブが現在のシステム情報を取得するまでに数分かかる場合があります。返された Job ID を介して、関数 *GetJobStatusDetails* を使用し現在のジョブ状態を取得できます。

構文

```
StartSystemInfoRetrieval ServerId=<ServerIdentifier>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          ShutdownType=<int>
          ShutdownCommunity=<string>
          ShutdownUsername=<string>
          ShutdownPassword=<string>
          FinalSeStAgentStatus=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerId
: サーバ ID を指定します。構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

DeploymentServer
: デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername
: 指定したデプロイメントサーバのユーザアカウント名を指定します。このパラメータは必須です。

DeploymentPassword
: デプロイメントサーバの指定ユーザアカウントに対してパスワードを指定します。このパラメータは必須です。

ShutdownType

サーバが動作中の場合の応答を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。ただしサーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI)
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
3 (SHUTDOWNTYPE_FORCE): ブレードタイプのサーバに対しては、マネージメントブレードによって強制的に電源を切断します。その他のタイプのサーバに対しては、ACPI で電源を切断します。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownCommunity

対象サーバで稼動している ServerView SNMP エージェントに対して送信するシャットダウン要求用の SNMP コミュニティを指定します。

ShutdownUsername

ServerView SNMP エージェントシャットダウンに対するユーザ名を指定します。
デフォルト値："

ShutdownPassword

前述のユーザに対するパスワードを指定します。
デフォルト値：" "

FinalSeStAgentStatus

ServerStart エージェントである Installation Manager エージェントの状態、およびインストール後のサーバの電源状態を指定します。
1 (AGENTSTOP_OFF): ServerStart Installation Manager エージェントは動作しません。サーバの電源を切断します。
2 (AGENTSTOP_REBOOT): ServerStart Installation Manager エージェントは動作しません。サーバを再起動します。
3 (AGENTSTOP_KEEPRUNNING): ServerStart Installation Manager エージェントは動作し続けます。
デフォルト値：1 (AGENTSTOP_OFF)

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

ShutdownType が 2 の場合、パラメータ *ShutdownType*、*ShutdownUsername*、*ShutdownPassword* を指定してください。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

JobID

詳細を取得するために *GetJobStatusDetails* に対して使用する、ユニークな ID です。

5.4.4 GetSystemInfoResult

GetSystemInfoResult 関数は、*StartSystemInfoRetrieval* により取得したシステム情報を XML 文書形式で提供します。現在のシステム情報を取得するには、事前に *StartSystemInfoRetrieval* を実行する必要があります。

構文

```
GetSystemInfoResult ServerId=<ServerIdentifier>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerId
: サーバ ID を指定します。構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

ModificationDate
: 指定したサーバに関するシステム情報の最終更新日付です（1970 年からの秒数）。

SystemInfo
: 指定したサーバに関する XML 形式のシステム詳細情報です。

5.4.5 RdDeplServerList

デプロイメントテーブルにある使用可能なサーバすべての一覧を取得するため、Job API は関数 *RdDeplServerList* をサポートしています。NodeList API がリリースされない限り、この関数はサポートされます。

構文

```
RdDeplServerList RDServer=<string>
          UserKey=<string>
```

パラメータ

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

DeplServerList

DeplServer タイプの構造体の配列です。

ServerView データベースは、*RdDeplServerList* を呼び出すたびにアクセスするわけではありません。ServerView データベースの変更が Deployment Manager で更新され *RdDeplServerList* によって返されるまでに最大 2 分の遅延が発生する場合があります。

戻りリストのレイアウト

デプロイメントサーバの戻りリストのその詳細のレイアウトは次のとおりです。意味を持つフィールドはサーバの種類によって異なります。

```
ServerName=<string>
ServerType=<int>
SV_SystemName=<string>
SV_IPAddress=<string>
SV_IPAddress=<string>
SV_Status=<string>
SystemType=<string>
Administrator=<string>
Location=<string>
MgmtServerName=<string>
MgmtIPAddress=<string>
MgmtSlot=<int>
ServerModel=<string>
CurrentImage=<string>
NewImage=<string>
DeploymentStatus=<string>
MacAddress1=<string>
IPAddress1=<string>
NetMask1=<string>
Gateway1=<string>
MacAddress2=<string>
IPAddress2=<string>
NetMask2=<string>
Gateway2=<string>
MacAddress3=<string>
IPAddress3=<string>
NetMask3=<string>
Gateway3=<string>
MacAddress4=<string>
IPAddress4=<string>
NetMask4=<string>
Gateway4=<string>
UserKeyInfo=<string>
```

ServerName

デプロイメント構成で指定したとおりのサーバ名を表示します。後述の *SV_SystemName*（OS ホスト名) と同じ名前である必要はありません。

ServerType

サーバの種類を表示します。
Server (ServerView)
BladeServer (ServerView)
Blade (ServerView)
BareServer (Deployment Manager)
サーバブレードをサーバとして ServerView データベースに入力した場合、同じホスト名の項目が 2 回表示されます。

SV_SystemName

ServerView データベースからのシステム名を表示します。すでにサーバ上に稼動している OS がインストールされている場合は、これは OS ホスト名になります。

SV_IPAddress

ServerView データベースの関連項目で指定したサーバの IP アドレスを表示します。

SV_Status

ServerView データベース内でのサーバの状態を表示します。
OK
Degraded
Fault
snmpOK
RSB_Mode
Not Manageable
Unknown
Ping

SystemType

サーバの種類に関する詳細を表示します。

Administrator

サーバの管理者を表示します。

Location

サーバのロケーションに関する詳細説明を表示します。

MgmtServerName

指定したサーバが属するブレードサーバのマネージメントブレードのホスト名を表示します。

MgmtIPAddress

指定したサーバが属する（*ServerType* が *Blade* の場合）ブレードサーバからマネージメントブレードの *SV_IPAddress* を表示します。*ServerType* が *Server* または *BareServer* で、さらに Ipmi でサーバを管理できる場合は、BMC の IP アドレスを表示します。

MgmtSlot

ブレードサーバ内にあるこのブレードのスロット ID を表示します（1 から 20）。

ServerModel

富士通サーバモデルの標準名称を表示します（例：D1710 は、詳細名称の S26361-D1710-A1 から取られました）。

CurrentImage

現在ロードされているイメージ名を表示します。デプロイメント状態が *cloned* の場合、デプロイメントテーブルの *MasterImageReference* は *CurrentImage*（*NewImage* は ""）になります。デプロイメント状態が *cloned* 以外の場合、*MasterImageReference* は *NewImage*（*CurrentImage* には最後にクローニングされたイメージの名前が含まれます）です。

NewImage

サーバが所属する（クローニングジョブが開始されるとすぐにサーバはグループのメンバになります）グループのイメージ名を表示します。

DeploymentStatus

デプロイメント状態を表示します。
"" (空白)
unknown
not cloned
cloning
cloned
installing
installed

MacAddress1

LAN ポート（インタフェース）1 の物理アドレスを表示します。

IPAddress1

LAN ポート（インタフェース）の現在の IP アドレスを表示します。DHCP を使用する場合、*IPAddress1* は空白です。

NetMask1

LAN ポート（インタフェース）1 の現在のネットマスクを表示します。DHCP を使用する場合、*NetMask1* は空白です。

Gateway1
: LAN ポート（インタフェース）1 の現在のゲートウェイを表示します。DHCP を使用する場合、*Gateway1* は空白です。

MacAddress2
: LAN ポート（インタフェース）2 の物理アドレスを表示します。

IPAddress2
: LAN ポート（インタフェース）2 の現在の IP アドレスを表示します。DHCP を使用する場合、*IPAddress2* は空白です。

NetMask2
: LAN ポート（インタフェース）2 の現在のネットマスクを表示します。DHCP を使用する場合、*NetMask2* は空白です。

Gateway2
: LAN ポート（インタフェース）2 の現在のゲートウェイを表示します。DHCP を使用する場合、*Gateway2* は空白です。

MacAddress3
: LAN ポート（インタフェース）3 の物理アドレスを表示します。

IPAddress3
: LAN ポート（インタフェース）3 の現在の IP アドレスを表示します。DHCP を使用する場合（将来）、*IPAddress3* は空白です。

NetMask3
: LAN ポート（インタフェース）3 の現在のネットマスクを表示します。DHCP を使用する場合（将来）、*NetMask3* は空白です。

Gateway3
: LAN ポート（インタフェース）3 の現在のゲートウェイを表示します。DHCP を使用する場合、*Gateway3* は空白です。

MacAddress4
: LAN ポート（インタフェース）4 の物理アドレスを表示します。

IPAddress4
: LAN ポート（インタフェース）の現在の IP アドレスを表示します。DHCP を使用する場合、*IPAddress4* は空白です。

NetMask4
: LAN ポート（インタフェース）4 の現在のネットマスクを表示します。DHCP を使用する場合、*NetMask4* は空白です。

Gateway4
: LAN ポート（インタフェース）4 の現在のゲートウェイを表示します。DHCP を使用する場合、*Gateway4* は空白です。

UserKeyInfo

サーバが未予約、自身により予約済み、外部ユーザにより予約済みのいずれかであるかを表示します。使用可能な値は次のとおりです。
"" (空白)
userkey_none
userkey_own
userkey_foreign

NodeList API がリリースされない限り、この Job API 関数はサポートされます。NodeList API がリリースされるとこの Job API 関数は廃止され、Job API の今後のバージョンではサポートされなくなります。

インタフェースの数は現在 4 つに制限されています。代わりに *ExportDeploymentConfiguration*（57 ページの「ExportDeploymentConfiguration」の項をご覧ください）を使って利用可能なインタフェースすべての情報をエクスポートできます。

5.4.6 ImportDeploymentConfiguration

Job API では関数 *ImportDeploymentConfiguration* を使い、1 つ以上のデプロイメント構成を XML 文書形式でインポートします。このデプロイメント構成は *ExportDeploymentConfiguration* を実行した際に作成されたものです。

構文

```
ImportDeploymentConfiguration DeploymentConfiguration=<XML>
          Params=<ArrayOf_ParameterData>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

DeploymentConfiguration
: XML 文書形式のデプロイメントテーブルを指定します。

Params
: *ParameterData* タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は 55 ページの " 使用可能なパラメータの表 " をご覧ください。また、パラメータの構造体については、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

使用可能なパラメータの表

この表では、パラメータの配列で指定できるパラメータを表示します。

AddNonExistingBareServers
: サーバリストに存在しないベアサーバのデプロイメント構成のインポートを有効または無効にします。
0 (OFF): サーバリストに存在しないベアサーバをインポートしません。
1 (ON): サーバリストに存在しないベアサーバをインポートします。
デフォルト値：0 (OFF)

MatchServerId

所定のデプロイメント構成に格納されているサーバ ID とサーバリストにあるサーバ ID が一致するサーバのデプロイメント構成のインポートを有効、または無効にします。
0 (OFF): サーバリストにあるサーバ ID と一致するサーバ ID を持つサーバのデプロイメント構成をインポートしません。
1 (ON): サーバリストにあるサーバ ID と一致するサーバ ID を持つサーバのデプロイメント構成をインポートします。
デフォルト値：0 (OFF)

MatchSlotId

デプロイメント構成に格納されているスロット番号とサーバリストにあるスロット番号が一致するサーバブレードのデプロイメント構成のインポートを有効、または無効にします。
0 (OFF): サーバリストにあるスロット番号と一致するスロット番号を持つサーバブレードのデプロイメント構成をインポートしません。
1 (ON): サーバリストにあるスロット番号と一致するスロット番号を持つサーバブレードのデプロイメント構成をインポートします。
デフォルト値：0 (OFF)

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.4.7 ExportDeploymentConfiguration

関数 *ExportDeploymentConfiguration* は所定のサーバのデプロイメント構成を XML 文書形式でエクスポートします。指定したサーバがマネージメントブレードの場合、既存ブレードのデプロイメント構成もすべて返されます。

構文

```
ExportDeploymentConfiguration
          ServerIdList=<Array of ServerIdentifier>
          Params=<ArrayOf_ParameterData>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerIdList

サーバリストにあるサーバの MAC アドレス、IP アドレスまたはホスト名を含む *ServerIdentifier* タイプの構造体の配列を指定します。この *ServerIdentifier* 構造体の定義については、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

Params

ParameterData タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は、58 ページの " 使用可能なパラメータの表 " をご覧ください。またパラメータの構造体については、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

使用可能なパラメータの表

PasswordDecoded

エクスポートされたデプロイメント構成に含まれるパスワードの復号を有効または無効にします。
0 (OFF): 暗号化されたパスワード
1 (ON): 復号化されたパスワード
デフォルト値：0 (OFF)

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。最後に実行した際の診断番号です。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

DeploymentConfiguration

XML 形式でエクスポートされたデプロイメント構成です。

5.4.8 SetUserKey

ユーザがサーバを予約する場合、Job API 関数 *SetUserKey* を使って、指定したサーバまたはサーバのリストを予約済みとマークします。

サーバの予約は、指定した *UserKey* のみに適用されますのでご注意ください。*UserKey* をなくしてしまうと、その *UserKey* を使って行ったサーバの予約を取り消すことができなくなります。

構文

```
SetUserKey ServerList=<string>
           RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerList

予約する様々なサーバの MAC アドレスのリストを指定します。例：00-C0-12-34-5F-67、00-C0-A4-12-34-23

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.4.9 RemoveUserKey

この関数は、指定したサーバの呼び出し元ユーザから利用可能なユーザ予約を削除します。

構文

```
SetUserKey ServerList=<string>
           RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerList

サーバの MAC アドレスのリストを指定します。例：00-C0-12-343-5F-67、00-C0-A4-12-34-23.

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.5 電源管理

Job API では、サーバの電源状態を制御できます。次に示す電源管理の関数を使って、対応するマネージメントブレード経由でサーバブレードを、そして IPMI 経由でスタンドアロンサーバの電源切断や電源投入を直接行うことができます。

5.5.1 SetPower

SetPower には、BMC IP address や BMC user name のような特定のパラメータは必要ありません。電源の投入または切断方法に関する具体的な情報を構成し、ServerView Deployment Manager サービスに対してその情報を設定する必要があります。

構文

```
SetPower ServerId=<ServerIdentifer>
           PowerStatus=<int>
           Params=<ArrayOf_ParameterData>
           RDServer=<string>
         UserKey=<string>
```

パラメータ

ServerId

サーバのサーバ ID を指定します。構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

PowerStatus

電源スイッチの状態を指定します。
0 (POWERSTATUS_OFF)
1 (POWERSTATUS_ON)

Params

ParameterData タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は、62 ページの " サーバの電源を切断するために使用できるパラメータの表 " と 63 ページの " サーバの電源を投入するために使用できるパラメータの表 " をご覧ください。またパラメータの構造体については 27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

サーバの電源を切断するために使用できるパラメータの表

表は指定したサーバの電源状態を電源切断に設定するパラメータを表示します。このパラメータはパラメータの配列で指定できます。

ShutdownType

シャットダウンの種類を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。サーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI)
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
3 (SHUTDOWNTYPE_FORCE): ブレードタイプのサーバに対しては、マネージメントブレードによって強制的に電源を切断します。その他のタイプのサーバに対しては、ACPI で電源を切断します。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownIp

ServerView エージェントシャットダウン用のホスト名または IP アドレスを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

ShutdownCommunity

シャットダウン対象のシステムで実行される ServerView エージェントシャットダウンに対して SNMP アクセス用コミュニティを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

ShutdownUsername

ServerView エージェントシャットダウンのユーザアカウント名を指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

ShutdownPassword

ServerView エージェントシャットダウン用ユーザに対するパスワードを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

Timeout

ServerView エージェント、マネージメントブレード、または BMC に対して発行される要求のタイムアウト値を秒単位で指定します。
ブレード用推奨値：6
BMC 用推奨値：3

Retries

シャットダウンの再試行回数を指定します。
ブレード用推奨値：2
BMC 用推奨値：1

サーバの電源を投入するために使用できるパラメータの表

表は指定したサーバの電源投入を設定するパラメータを表示します。このパラメータはパラメータの配列で指定できます。

BootMode

ブートモードを指定します。
0 (BOOTMODE_NORMAL)
1 (BOOTMODE_PXE)
デフォルト値：0 (BOOTMODE_NORMAL)

ShutdownType

シャットダウンの種類を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。ただしサーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI)
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
3 (SHUTDOWNTYPE_FORCE): ブレードタイプのサーバに対しては、マネージメントブレードによって強制的に電源を切断します。その他のタイプのサーバに対しては、ACPI で電源を切断します。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownIp

ServerView エージェントシャットダウン用のホスト名または IP アドレスを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

ShutdownCommunity

シャットダウン対象のシステムで実行される ServerView エージェントシャットダウンに対して SNMP アクセス用コミュニティを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

ShutdownPassword

ServerView エージェントシャットダウン用ユーザに対するパスワードを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。

Timeout

ServerView エージェント、マネージメントブレード、または BMC に対して発行される要求のタイムアウト値を秒単位で指定します。
ブレード用推奨値：6
BMC 用推奨値：3

Retries

シャットダウンの再試行回数を指定します。
ブレード用推奨値：2
BMC 用推奨値：1

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.5.2 SetPXEBootImage

関数 SetPXEBootImage を使って PXE クライアント用のブートイメージを PXE サービスへ登録する、または登録を解除することができます。

構文

```
SetPXEBootImage ServerId=<ServerIdentifer>
           BootImagePath=<string>
           BootImageType=<int>
           Params=<ArrayOf_ParameterData>
           DeploymentServer
           DeploymentUsername
           DeploymentPassword
           RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerId

ServerIdentifier タイプの構造体でサーバを指定します。MAC アドレスでサーバを指定する場合、PXE クライアントは所定の MAC アドレスによりそのまま登録されます。サーバを IP アドレス、またはホスト名で指定する場合、PXE クライアントは *PXEBootInterface* としてデプロイメント構成に登録された MAC アドレスにより登録されます。*ServerIdentifier* の構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

BootImagePath

ブートイメージ名を含む、tftp サーバのルートディレクトリへの相対パスを指定します。DOS イメージでは、サーバのルートディレクトリへの相対パスか、またはファイルシステム内の絶対パスのいずれかになります。絶対パスの場合、ファイルは自動的に tftp サーバの一時ディレクトリへコピーされます。

BootImageType

ブートイメージの種類を指定します。
0 (BOOTIMAGETYPE_BOOTSTRAPIMAGE)
1 (BOOTIMAGETYPE_DOSIMAGE)
デフォルト値：0 (BOOTIMAGETYPE_BOOTSTRAPIMAGE)

Params

ParameterData タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は、67 ページの "PXE クライアント用のブートイメージを登録するために使用可能なパラメータの表 " と 67 ページの "PXE クライアント用ブートイメージの登録を解除するために使用可能なパラメータの表 " をご覧ください。またパラメータの構造体については、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

DeploymentServer

PXE サービスが稼動しているデプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。(このリリースでは、RDServer と同じでなければなりません)。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウントを指定します。このパラメータは必須です。

DeploymentPassword

デプロイメントサーバ上で指定したユーザアカウントに対するパスワードを指定します。このパラメータは必須です。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

Params 構造体でパラメータを指定しない場合、デフォルトで非持続的登録が行われます。

PXE クライアント用のブートイメージを登録するために使用可能なパラメータの表

表は、パラメータの配列で指定できるパラメータを表示します。

RegisterMode

指定した PXE ブートイメージの持続的または非持続的登録を指定します。
0 (REGISTER_NONPERSISTENT): 次回の PXE ブート用にサーバを PXE サービスに登録します。1 回 PXE ブートを実行した後、サーバは自動的に PXE サービスの登録から解除されます（ブートイメージは登録されたままです)。
1 (REGISTER_PERSISTENT): サーバを持続的に PXE サービスに登録します。
デフォルト値：0 (REGISTER_NONPERSISTENT)

PXE クライアント用ブートイメージの登録を解除するために使用可能なパラメータの表

表は、パラメータの配列で指定できるパラメータを表示します。

UnregisterMode

登録解除モードを指定します。
1 (UNREGISTER_BOOTIMAGE_AND_CLIENT): 所定のブートイメージからサーバの登録を解除し、ブートイメージ自身の登録も解除します（このブートイメージに他の PXE クライアントが登録されていない場合)。サーバが登録されていない場合（サーバが REGISTER_NONPERSISTENT モードで登録されていて、PXE ブートがすでに実行されているときに発生する場合があります)、エラーを返さずにブートイメージの登録が解除されます（このブートイメージに他のクライアントが登録されていない場合)。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.5.3 GetPower

関数 GetPower を使ってサーバの現在の電源状態を確認します。

構文

```
GetPower ServerId=<ServerIdentifer>
           Params=<ArrayOf_ParameterData>
           RDServer=<string>
         UserKey=<string>
```

パラメータ

ServerId

ServerIdentifier タイプの構造体でサーバの MAC アドレスを指定します。詳細は 26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

Params

ParameterData タイプの構造体の配列を指定します。使用可能なパラメータに関する詳細は、69 ページの " 使用可能なパラメータの表 " をご覧ください。またパラメータの構造体については、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

使用可能なパラメータの表

表は、パラメータの配列で指定できるパラメータを表示します。

Timeout

マネージメントブレードに対して発行される要求のタイムアウト値を秒単位で指定します。
推奨値：6

Retries

SNMP 再試行回数を指定します。
推奨値：2

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

PowerStatus

現在の電源状態を返します。
0 (POWERSTATUS_OFF)
1 (POWERSTATUS_ON)

5.5.4 GetBootMode

関数 *GetBootMode* を使って、MAC アドレスを介して指定したサーバの現在のブートモードを取得できます。Job API はシステムタイプ情報に基づいて、関数がマネージメントブレードまたは IPMI を利用した BMC により管理されているかどうかを直接確認します。

構文

```
GetBootMode ServerId=<ServerIdentifer>
            RDServer=<string>
           UserKey=<string>
```

パラメータ

ServerId

ServerIdentifier タイプの構造体でサーバの MAC アドレスを指定します。詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

BootMode

ブートモードを返します。
0 (BOOTMODE_NORMAL)
1 (BOOTMODE_PXE)

5.6 リポジトリ管理

リポジトリとは、Deployment Service や ServerView Deployment Manager によりアクセスできるコンテナ（Windows サーバでは共有ディレクトリ）のようなものです。リポジトリは複数のディレクトリやサブディレクトリに分割できます。しかし原則として、Deployment Manager では、リポジトリとそのディレクトリはイメージを格納するために使用されます。

イメージは Deployment Manager と Job API 内において注目されるものです。基本的に IT におけるイメージとは、ハードディスク、パーティション、または CD のコピーのようなものです。Job API 内においてイメージ作成とは完全にインストールされ構成されたシステムからイメージを作成することを意味します。すなわち、イメージには参照システム（例：ハードディスク、付加記憶領域、パーティションの構成、OS のデータすべてなど）の構成情報が含まれます。したがって、イメージは参照イメージとして使用され、複数の対象システムをデプロイメントします。

新しい Job API では、*CreateImage* という新しい汎用関数を使用できます。この関数を使って、サポートされている種類すべてのイメージを作成することができます。すなわち、*CreateImage* では、snapshot イメージも作成できます。

原則として、様々なイメージの種類が格納されるため、格納先のリポジトリもそれぞれ異なります。次では、現在の Job API がサポートするリポジトリの種類をすべて記載します。

– クローニングイメージ
  リポジトリは、Job API または Deployment Manager を介して作成されたディスクイメージファイルすべてを収容します。またこれらリポジトリはクローニングを実行するために使用されます。

– Boot Images
  ブートイメージのリポジトリは、汎用システムを準備するために作成された PXE ブートイメージを収容するディレクトリです。

– インストール構成
  インストール構成ファイルのリポジトリは、インストール構成ファイルが格納されているサブディレクトリを持つディレクトリです。このインストール構成ファイルは Installation Manager を使って作成する必要があります。

– デプロイメントテーブル
  デプロイメントテーブルのリポジトリは、Deployment Manager フロントエンドの関数 *Export Deployment Configuration* を使ってエクスポートされた XML ファイルを収容します。

例

リポジトリ内の各イメージオブジェクトは同一ファイルセットを表しますが、それぞれの拡張子は異なります。

– バイナリクローニングイメージファイル (*.img*)

  このファイルは、ハードディスク（ネットのどこかにある格納場所のURL）のコピーを収容します。

– イメージ文書 (*.txt*)

  このファイルは、イメージを説明するために使用できます。例：イメージ作成中に作成されるテキスト文書の参照

– ハードウェア /OS パラメータファイル (*.cfg*)

  この情報は、既存イメージの変更と割り当てられた対象 HW に対する互換性の確認のために使用します。

– リモート Mass Installation ファイル (*.xml*)

  この構成ファイルは Installation Manager ウィザード（例：準備モード）により作成されます。

次の項では、新しいリポジトリの追加、不要なリポジトリの削除、既知リポジトリの取得に使用されるリポジトリ関数を記載しています。さらに、Job API ではリポジトリに収容されたイメージを扱ったり、新しいイメージを作成したりするための関数を使用できます。

5.6.1 AddRepository

リポジトリを追加するためには、Job API でリポジトリを識別するために使用するシンボリック名とパス名を指定してください。一般的に、パス名とは、共有ディレクトリへのネットワークパス（UNC 表記法）のことです。パス名は、ネットワークパスではなくローカルディレクトリへの絶対パスでも可能です。しかし、その場合、ServerView Deployment Manager サービスと Deployment Service が同じマシンにインストールされている必要があります。これらサービスがインストールされていない場合、UNC 表記法による共有へのネットワークパス *\\server_name\share_name* でなければなりません。

構文

```
AddRepository RepositoryName=<string>
              RepositoryType=<int>
              NetworkPath=<string>
              RDServer=<string>
             UserKey=<string>
```

パラメータ

RepositoryName
: 新しいリポジトリの名前を指定します。

RepositoryType
: 新しいリポジトリの種類を指定します。
1 (REPOSITORYTYPE_CLONING_IMAGE)
2 (REPOSITORYTYPE_BOOT_IMAGE)
3 (REPOSITORYTYPE_CONFIG_FILE)
4 (REPOSITORYTYPE_DEPLOYMENT_TABLE)
デフォルト値：1 (REPOSITORYTYPE_CLONING_IMAGE)

NetworkPath
: 後で該当するイメージファイルを収容する場所の完全修飾ネットワークパスを UNC 形式で指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値：""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.6.2 RemoveRepository

関数 RemoveRepository を使って、不要なリポジトリを削除します。したがって、既存のリポジトリ、デプロイメントサーバ、およびユーザアカウントを識別するユーザキーのシンボリック名を指定してください。

構文

```
RemoveRepository RepositoryName=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

RepositoryName
: 削除するリポジトリ名を指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

この関数を使うと、イメージオブジェクトではなくリポジトリー覧のエントリが削除されます。

5.6.3 GetRepositoryList

関数 *GetRepositoryList* は既知リポジトリすべての情報を返します。既存リポジトリの種類をそれぞれ表示するためには、該当するリポジトリの種類、または全種類に対しては 0（ALL) を選びます。

構文

```
GetRepositoryList RepositoryType=<ServerIdentifer>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

RepositoryType

情報を取得するリポジトリの種類を指定します。
0 (ALL)
1 (REPOSITORYTYPE_CLONING_IMAGE)
2 (REPOSITORYTYPE_BOOT_IMAGE)
3 (REPOSITORYTYPE_CONFIG_FILE)
4 (REPOSITORYTYPE_DEPLOYMENT_TABLE)
デフォルト値：0 (ALL)

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

RepositoryList
: *RepositoryData* タイプの構造体の配列を返します。

戻りリストのレイアウト

リポジトリの戻りリストとその詳細のレイアウトは次のとおりです。

```
RepositoryName=<string> RepositoryType=<int>
NetworkPath=<string>
```

RepositoryName
: リポジトリの名前を表示します。

RepositoryType
: リポジトリの種類を表示します。
1 (REPOSITORYTYPE_CLONING_IMAGE)
2 (REPOSITORYTYPE_BOOT_IMAGE)
3 (REPOSITORYTYPE_CONFIG_FILE)
4 (REPOSITORYTYPE_DEPLOYMENT_TABLE)

NetworkPath
: ネットワークパスを UNC 形式で表示します。

5.6.4 CreateImage

関数 *CreateImage* を使ってクローニングイメージとスナップショットイメージを作成します。これらのイメージを使って複数のサーバ（デプロイメントテーブルに基づいて個別化されたもの）をデプロイメントします。パラメータ *ImageType* は生成されたイメージの種類を選択します。パラメータ Params にはイメージ作成パラメータのリストが含まれ、*ImageType* に応じて解釈されます。*CreateImage* は、イメージの作成を開始し、Job ID を返します。この Job ID を以降の *GetJobStatusDetails* の呼び出し入力項目として使用できます。

構文

```
CreateImage ServerId=<int>
            ImagePath=<string>
            ImageType=<int>
            InfoText=<string>
            Params=<ArrayOf_ParameterData>
            DeploymentServer=<string>
            DeploymentUsername=<string>
            DeploymentPassword=<string>
            RDServer=<string>
            UserKey=<string>
```

パラメータ

ServerId

ServerIdentifier タイプの構造体で対象サーバブレードの MAC アドレスを指定します。詳細は 26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

ImagePath

イメージの名前を含む完全修飾パスを指定します。

ImageType

イメージの種類を指定します。
1 (IMAGETYPE_CLONING)
2 (IMAGETYPE_SNAPSHOT)

InfoText

イメージと共に保存する情報を指定します。情報は *.txt* ファイルで保存されます。

Params

ParameterData タイプの構造体の配列を指定します。パラメータの構造体に関して詳しくは、27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。使用可能なパラメータについては 81 ページの " クローニングイメージを作成するために使用可能なパラメータの表 " および 84 ページの " スナップショットイメージを作成するために使用可能なパラメータの表 " をご覧ください。

DeploymentServer

デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウント名を指定します。

DeploymentPassword

デプロイメントサーバで指定したユーザアカウントに対するパスワードを指定します。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

i *ShutdownType* が 3 で、IPMI を搭載していないサーバを指定した場合、シャットダウンを手動で行う必要があるため、この関数は正常に実行したと返します。

クローニングイメージを作成するために使用可能なパラメータの表

次のパラメータ表では、*IMAGETYPE_CLONING* タイプのイメージを作成するために Params の配列で指定する固有の名前と値を説明します。

SaveMode

ハードディスクのイメージ作成中に使用する保存モードを指定します。
0 (SAVEMODE_FILESYSTEM_DEPENDENT): パーティション内で使用されているセクタのみを保存します。
1 (SAVEMODE_FILESYSTEM_INDEPENDENT): パーティション内のすべてのセクタを保存します。
2 (SAVEMODE_RAW): ディスクすべてを保存します。
デフォルト値：0 (SAVEMODE_FILESYSTEM_DEPENDENT)

ShutdownType

サーバが動作している場合の応答を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。ただし、サーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI):
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownUsername

ServerView エージェントシャットダウンに対するユーザアカウント名を指定します。*ShutdownType* が 2 の場合、このパラメータは必須です。
デフォルト値："

ShutdownPassword

前述のユーザに対するパスワードを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。
デフォルト値："

FinalPowerStatus

イメージ生成後のサーバの電源状態を指定します。
0 (FINALPOWERSTATUS_OFF): サーバの電源は切断されています。
1 (FINALPOWERSTATUS_ON): インストールされている OS が起動しています。
このパラメータを使って、イメージ生成後、サーバの OS を起動するか、またはサーバの電源を切断するかを選択できます。
デフォルト値：0 (FINALPOWERSTATUS_OFF)

OS

サポートする OS を指定します。
Windows:Windows2000 または Windows2003 システムからクローニングイメージを作成します。
Windows2008: Windows2008 システムからクローニングイメージを作成します。
Linux: Linux システムからクローニングイメージを作成します。
このパラメータは必須です。

LinuxConfigLocation

イメージ作成を行うシステムの */etc* ディレクトリの場所を指定します。
このパラメータは、Linux を OS として選択した場合のみ適用され、必須です。

WindowsAdminName

OS WINDOWS（Windows 依存）に対して管理者権限を持つログイン名を指定します。このパラメータは、Windows または Windows2008 を OS として選択した場合のみ適用されます。
デフォルト値："
これは参照システムのローカル管理者アカウントである必要があります。
デプロイメントサーバまたは参照システムがドメインに所属している場合、<*hostname*>*administrator* または <*IP address*>*administrator* の形式でこのパラメータを指定してください。

WindowsAdminPassword

前述の管理者ユーザに対するパスワードを指定します。
このパラメータは Windows または Windows2008 を OS として選択している場合のみ適用されます。
デフォルト値："

WindowsProductID

参照システムの ProductID を指定します。
このパラメータは、Windows（必須）または Windows2008（オプション）を OS として選択した場合のみ適用されます。
デフォルト値："

LogicalDisk

論理ディスクを指定します。
デフォルト値：0

Verify

バックアップを行う前にファイルシステムを確認します。
0: (OFF)
1: (ON)
デフォルト値：1 (ON)

Compress

圧縮を実行します。
0: (OFF)
1: (ON)
デフォルト値：1 (ON)

UnknownFail

未知のファイルシステムが検出された場合、SaveMode= SAVEMODE_FILESYSTEM_DEPENDENT と合わせて応答を指定します。
0 (OFF): 関数処理を続行します。
1 (ON): 関数処理はエラーになり失敗しました。
デフォルト値：0 (OFF)

FastMode

Linux のクローニングイメージを作成するために使用します。ext2/ext3 ファイルシステムのイメージ生成の最適化を指定します。
0 (OFF): イメージ高速生成を無効にします。
1 (ON): イメージ高速生成を有効にします。
最適化されて作成されたイメージは ExpandLastPartition を使ってクローニングできません（Deploy をご覧ください）。
デフォルト値：0 (OFF)

ServerAccessData

このパラメータは、Windows または Windows2008 を OS として選択した場合のみ適用されます。このパラメータには、システムの準備中に Windows システムへアクセスするために使用する IP アドレスまたはホスト名（完全修飾名が許可されています）を含んでいる場合があります。このパラメータが空白の場合、デプロイメント構成のサーバ名が使用されます。
デフォルト値："

InstallationLocale

Windows2008 のみで使用できます。このパラメータは、"Fujitsu OEM" DVD ("Multi Language" DVD) から参照システムをインストールした場合に使用します。"Microsoft" DVD に対してこのパラメータを使用しないでください。
デフォルト値："
使用可能な値：
English (United States)
German (Germany)
Japanese (Japan)
French (France)
Spanish (Spain)
Italian (Italy)
Dutch (Netherlands)
Portuguese (Portugal)
Portuguese (Brazil)
Czech (Czech Republic)
Polish (Poland)
Russian (Russia)
Swedish (Sweden)
Hungarian (Hungary)
Turkish (Turkey)
Korean (Korea)
Chinese (PRC)

スナップショットイメージを作成するために使用可能なパラメータの表

次のパラメーター覧表では、IMAGETYPE_SNAPSHOT タイプのイメージを作成するために *Params* の配列で指定する固有の名前と値を説明します。

SaveMode

ハードディスクのイメージ作成中に使用する保存モードを指定します。
0 (SAVEMODE_FILESYSTEM_DEPENDENT): パーティション内で使用されているセクタのみを保存します。
1 (SAVEMODE_FILESYSTEM_INDEPENDENT): パーティション内のすべてのセクタを保存します。
2 (SAVEMODE_RAW): ディスクすべてを保存します。
デフォルト値：0 (SAVEMODE_FILESYSTEM_DEPENDENT)

ShutdownType

サーバが動作している場合の応答を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。ただし、サーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI):
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownUsername

ServerView エージェントシャットダウンに対するユーザアカウント名を指定します。
ShutdownType が 2(SHUTDOWNTYPE_GRACEFUL) の場合、このパラメータは必須です。
デフォルト値："

ShutdownPassword

前述のユーザに対するパスワードを指定します。
ShutdownType が 2 の場合、このパラメータは必須です。
デフォルト値："

FinalPowerStatus

イメージ生成後のサーバの電源状態を指定します。
0 (FINALPOWERSTATUS_OFF): サーバの電源が切断されています。
1 (FINALPOWERSTATUS_ON): インストールされている OS が起動されます。
このパラメータを使って、イメージ生成後、サーバ OS システムを起動するか、またはサーバの電源を切断するかを選択します。
デフォルト値：0 (FINALPOWERSTATUS_OFF)

LogicalDisk

論理ドライブを指定します。
デフォルト値：0

Verify

バックアップを行う前にファイルシステムを確認します。
0: (OFF)
1: (ON)
デフォルト値：1 (ON)

Compress

圧縮を実行します。
0: (OFF)
1: (ON)
デフォルト値：1 (ON)

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合は、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合は、*ErrorString* は空白です。

JobID

詳細を取得するために *GetJobStatusDetails* に対して使用する、ユニークな ID です。

5.6.5 DeleteImage

DeleteImage 関数は指定したイメージファイルを削除します。

構文

```
DeleteImage Imagefilename=<string>
          RepositoryName=<string>
          SubPath=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

Imagefilename
: 削除するイメージの名前を指定します。

RepositoryName
: イメージが含まれているリポジトリ名を指定します。

SubPath
: 削除するイメージが含まれているリポジトリへの完全修飾サブパスを指定します。
デフォルト値："

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合は、*ErrorString* は空白です。

5.6.6 GetRepositoryContents

関数 GetRepositoryContents を使って、指定したリポジトリまたは指定したタイプのすべてのリポジトリで検出されたイメージファイル名を一覧表示します。

構文

```
GetRepositoryContents RepositoryName=<string>
          RepositoryType=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

RepositoryName

リポジトリ名を指定します。*RepositoryName* を指定しない場合、指定した種類のリポジトリにある内容がすべて表示されます。
デフォルト値："

RepositoryType

リポジトリの種類を指定します。
0 (ALL)
1 (REPOSITORYTYPE_CLONING_IMAGE)
2 (REPOSITORYTYPE_BOOT_IMAGE)
3 (REPOSITORYTYPE_CONFIG_FILE)
4 (REPOSITORYTYPE_DEPLOYMENT_TABLE)
デフォルト値：0 (ALL)

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合は、*ErrorString* は空白です。

FilenameList

イメージファイル名の配列を返します。

5.6.7 ReadImagefilenameList

ReadImagefilenameList 関数は、既知の Deployment Manager リポジトリで検出されたイメージファイル名の配列を返します。

構文

```
ReadImagefilenameList RDServer=<string>
          UserKey=<string>
```

パラメータ

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合は、*ErrorString* は空白です。

FilenameList

イメージファイル名の配列を返します。

5.6.8 ReadImageInfoText

ReadImageInfoText 関数は、*CreateImage* 関数の呼び出し中に検出されたイメージ文書を取り出します。

構文

```
ReadImageInfoText ImagePath=<string>
          RDServer=<string>
          UserKey=<string>
```

Parameter

ImagePath
: イメージの名前を含む完全修飾パスを指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合は、*ErrorString* は空白です。

InfoText
: イメージ文書に戻ります。

5.6.9 GetImageProperties

GetImageProperties 関数は *CreateImage* 関数の呼び出し中に生成された、または SetImageProperties 関数によって定義されたイメージのプロパティを返します。

構文

```
GetImageProperties ImagePath=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ImagePath
: イメージの名前を含む完全修飾パスを指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

ImageProperties
: イメージの構成を XML 文書の形式で返します。

5.7 ディレクトリ管理

前述のとおり、既存のリポジトリには複数のディレクトリとサブディレクトリが存在します。この項では、リポジトリにあるディレクトリを管理するための Job API 関数すべてを説明します。

5.7.1 CreateDirectory

CreateDirectory 関数は指定したリポジトリに新しいディレクトリを作成します。

構文

```
CreateDirectory DirectoryName=<string>
          RepositoryName=<string>
          SubPath=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

DirectoryName
: 新しいディレクトリの名前を指定します。

RepositoryName
: 新しいディレクトリを収容するリポジトリの名前を指定します。

SubPath
: 指定したリポジトリに相対する、新しいディレクトリへの部分修飾パスを指定します。
デフォルト値："

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号を返します。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.7.2 DeleteDirectory

DeleteDirectory 関数は、所定のリポジトリ内の指定ディレクトリを完全に削除します。指定したディレクトリが空でない場合、すべてのファイル（イメージファイル）とサブディレクトリが最初に削除されます。

構文

```
DeleteDirectory DirectoryName=<string>
          RepositoryName=<string>
          SubPath=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

DirectoryName
: 削除するディレクトリの名前を指定します。

RepositoryName
: 指定したディレクトリを含むリポジトリの名前を指定します。

SubPath
: 指定のリポジトリに相対する、新しいディレクトリへの部分修飾パスを指定します。
デフォルト値："

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.7.3 GetDirectoryContents

GetDirectoryContents 関数は、指定したディレクトリにある既知のファイルとサブディレクトリの一覧を取得します。

構文

```
GetDirectoryContents DirectoryName=<string>
          RepositoryName=<string>
          SubPath=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

DirectoryName
: ディレクトリの名前を指定します。

RepositoryName
: リポジトリの名前を指定します。

SubPath
: 指定したリポジトリに相対するディレクトリへの部分修飾パスを指定します。
デフォルト値："

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

FileList

FileData 名の配列を返します。

戻りリストのレイアウト

```
Name=<string> Type=<int> Size=<int> ModificationDate=<long>
```

Name

ファイル名を表示します。

Type

ファイルの種類を表示します。
0 (FILETYPE_DIRECTORY)
1 (FILETYPE_INSTALLATION_CONFIG)
2 (FILETYPE_CLONING_IMAGE)
3 (FILETYPE_OTHER)

Size

ファイルのサイズを表示します。

ModificationDate

最終修正日を表示します。

5.8 グループ管理

大規模なデプロイメントにおいて、多数のサーバはほぼ同一の構成で同時並行にデプロイメントされます。同じ、または類似した構成のサーバを 1 つのグループにまとめると作業を最小限に抑えることができます。一般的にグループとは、共通でなおかつ個々のプロパティを持つグループメンバの集合体に名前をつけたものです。

Job API は 2 種類のグループの割り当てをサポートします。その 2 種類とは、1:N と N:N のグループプロパティ割り当てです。1:N プロパティ割り当ての場合、グループ内の各サーバに対して、指定したグループプロパティを割り当てます。一方、N:N 割り当ての場合、プロパティは *SetGroupMemberProperties* 関数を介してグループ内の各サーバに割り当てられます。また、指定したグループプロパティは考慮されません。

さらに、グループ内にもグループを作成することができます。こうして作成されたサブグループにより、より詳細なグループ化が可能になります。

5.8.1 CreateGroup

CreateGroup は指定したグループの種類（*GroupType*) に応じてグループを作成します。新しく作成されたグループは、グループの種類にかかわらず空であり、かつ、グループプロパティも存在しません。正常にグループの作成が完了すると、特有のグループ識別子（*GroupID*）が返されます。他のグループ関数でこの *GroupID* を使って、既存のグループを識別します。

構文

```
CreateGroup GroupName=<string>
          GroupType=<int>
          ParentGroupID=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupName
: 新しいグループの名前を指定します。

GroupType
: グループの種類を指定します。
2 (GROUPTYPE_CLONING)
3 (GROUPTYPE_INSTALLATION)
このパラメータは必須です。

ParentGroupID
: 新しいサブグループを作成するため既存のグループ識別子を指定します。
デフォルト値：0

現行バージョンの Job API では、*ParentGroupID* は指定できません。そのため、このパラメータは評価されません。したがって、現行バージョンではサブグループを作成することはできません。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値：""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

GroupID

新しいグループエントリ特有のグループ識別子を返します。

5.8.2 DeleteGroup

既存のグループを削除するためには、*DeleteGroup* 関数を使用します。指定したグループが空ではない場合、すべてのグループメンバがまず削除されます。グループメンバを削除することで、Job API の *RemoveGroupMember* 関数が呼び出されます。

構文

```
DeleteGroup GroupID=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupID
: 削除するグループエントリを指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.8.3 GetGroupList

GetGroupList 関数は、すべての既知のグループを一覧表示します。戻りリストの範囲は、特定の *GroupType* を 1 つ選択して制限することができます。

構文

```
GetGroupList GroupType=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupType

情報を取得するグループの種類を指定します。
0 (ALL)
2 (GROUPTYPE_CLONING)
3 (GROUPTYPE_INSTALLATION)
デフォルト値：0 (ALL)

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値：""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

GroupList

GroupData タイプの構造体の配列です。

戻りリストのレイアウト

```
GroupID=<int> GroupName=<string> GroupType=<int>
```

GroupID
: グループのグループ識別子を表示します。

GroupName
: グループの名前を表示します。

GroupType
: グループの種類を表示します。
2 (GROUPTYPE_CLONING)
3 (GROUPTYPE_INSTALLATION)

5.8.4 SetGroupProperties

SetGroupProperties 関数は、既存グループのプロパティを設定および修正します。したがって、*GetGroupList* 関数が返した有効なグループ識別子を識別する必要があります。

構文

```
SetGroupProperties GroupID=<int>
          GroupProperties=<ArrayOf_PropertyData>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupID
: 既存グループのグループ識別子を指定します。

GroupProperties
: *PropertyData* タイプの構造体の配列を指定します。この配列には、使用可能なグループのプロパティが含まれています。詳しくは、26 ページの "ArrayOf_PropertyData 構造体のレイアウト " をご覧ください。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行した場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合は、*ErrorString* は空白です。

グループプロパティのレイアウト

```
Name=<string> Value=<string>
```

Name
: グループプロパティの既知名を指定します。

Value
: 指定したプロパティの値を指定します。

クローニンググループプロパティの表

ImagePath
: イメージの名前を含む完全修飾パスを指定します。

5.8.5 GetGroupProperties

GetGroupMemberProperties 関数は、指定したメンバと *GroupID* に対するプロパティの一覧を返します。

構文

```
GetGroupProperties GroupID=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupID
: 既存グループの識別子を指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

GroupName
: グループの名前を返します。

GroupType
: グループの種類を返します。
2 (GROUPTYPE_CLONING)
3 (GROUPTYPE_INSTALLATION)

GroupProperties
: *PropertyData* タイプの構造体の配列を返します。詳しくは、106 ページの " グループプロパティのレイアウト " と 106 ページの " クローニンググループプロパティの表 " をご覧ください。

5.8.6 AddGroupMember

AddGroupMember 関数は、指定したグループへ新しいグループを追加します。現在の Job API バージョンでは、グループメンバは MAC アドレス、IP アドレス、またはホスト名（*MemberId*）によって識別されたサーバになることがあります。

構文

```
AddGroupMember MemberIdList=<ArrayOf_MemberIdentifier>
          GroupID=<int>
          RDServer=<string>
          UserKey=<string>
```

Parameter

MemberIdList
: 新しいグループメンバのリストの 1 つを指定します。*MemberIdentifier* 構造体についての詳細は、109 ページの "MemberIdentifier 構造体のレイアウト " をご覧ください。

GroupID
: 新しいメンバを追加するグループを指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

MemberIdList で指定したメンバのうち 1 つでも指定グループへ追加できない場合、関数にエラーが発生し、すべての指定メンバが追加されません。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

MemberIdentifier 構造体のレイアウト

メンバを識別するために識別子の種類を選択します（例：MAC アドレス、IP アドレス、ホスト名など）。次の構造体を使って、指定した種類でメンバを識別します。

```
Identifier=<string> Type=<int>
```

Identifier

メンバ特有の識別子を含む文字列を指定します。

Type

指定したメンバ識別子の種類を指定します。
0 (IDENTIFIERTYPE_MAC): ネットワークインタフェースの物理アドレス
1 (IDENTIFIERTPE_IP): サーバの IP アドレス。
これは、RdDeplServerList で返された SV_IPAddress フィールドである必要があります。
2 (IDENTIFIERTYPE_NAME): サーバのホスト 名。
これは、*RdDeplServerList* で返された *ServerName* フィールドである必要があります。

5.8.7 RemoveGroupMember

RemoveGroupMember 関数は、指定したグループメンバを所定のグループから削除します。グループ自身は削除されません。

構文

```
RemoveGroupMember MemberId=<MemberIdentifier>
          GroupID=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

MemberId
: 削除するグループメンバを指定します。
MemberIdentifier 構造体についての詳細は、109 ページの "MemberIdentifier 構造体のレイアウト " をご覧ください。

GroupID
: 前回 *CreateGroup* を呼び出した際に取得した *GroupID* を指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode
: 最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString
: *ErrorCode* のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.8.8 GetGroupMemberList

GetGroupMemberProperties 関数は、GroupID を使って、指定メンバと所定グループのプロパティを一覧表で返します。

構文

```
GetGroupMemberList GroupID=<int>
          MemberIdType=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupID

返されたメンバの既知グループを指定します。

MemberIdType

どの戻りメンバのメンバ識別子タイプを使用するかを指定します。
0 (IDENTIFIERTYPE_MAC): 最初のネットワークインタフェースの物理アドレスです。
1 (IDENTIFIERTYPE_IP): サーバの IP アドレスです。
2 (IDENTIFIERTYPE_NAME): サーバのホスト名です。指定した識別子が空の場合、別の識別子を返します。
デフォルト値：0 (IDENTIFIERTYPE_MAC)

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

MemberList

MemberIdentifier の構造体の配列を返します。

戻りリストのレイアウト

リポジトリの戻りリストとその詳細は次のレイアウトを使用します。

```
Identifier=<string> Type=<int>
```

Identifier

メンバ特有の識別子を含む文字列を表示します。

Type

パラメータ *MemberIdType* の設定に応じて、メンバ識別子の種類を表示します。
0 (IDENTIFIERTYPE_MAC): ネットワークインタフェースの物理アドレス。
1 (IDENTIFIERTYPE_IP): サーバの IP アドレス
2 (IDENTIFIERTYPE_NAME): サーバのホスト 名
3 (IDENTIFIERTYPE_GROUPID): グループ ID

5.8.9 SetGroupMemberProperties

SetGroupMemberProperties 関数を使って、グループメンバの各プロパティを設定します。*PropertyData* タイプ（名前 / 値のペア）を指定することによって、プロパティの任意リストを定義します。指定されたプロパティはこのグループ内のグループメンバ（サーバ）固有のものです。

構文

```
SetGroupMemberProperties MemberId=<MemberIdentifier>
          GroupID=<int>
          GroupMemberProperties=<ArrayOf_PropertyData>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

MemberId

グループメンバの識別子を指定します。*MemberIdentifier* についての詳細は 109 ページの "MemberIdentifier 構造体のレイアウト " をご覧ください。

GroupID

既存グループの識別子を指定します。

GroupMemberProperties

PropertyData タイプの構造体の配列を指定します。構造体についての詳細は 114 ページの " インストールグループメンバプロパティのレイアウト " をご覧ください。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

インストールグループメンバプロパティのレイアウト

```
Name=<string> Value=<string>
```

Name

グループメンバプロパティの既知名を指定します。

Value

指定したグループメンバの値を指定します。

グループメンバプロパティのインストール表

ConfigPath

Installation Manager で定義された構成名を含む完全修飾パスを指定します。

5.8.10 GetGroupMemberProperties

所定グループメンバのプロパティを取得するためには、*GetGroupMember* 関数を使ってグループメンバのプロパティを一覧表示します。

構文

```
GetGroupMemberProperties MemberId=<MemberIdentifier>
          GroupID=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

MemberId
: グループメンバの識別子を指定します。*MemberIdentifier* の構造体についての詳細は、109 ページの "MemberIdentifier 構造体のレイアウト "をご覧ください。

GroupID
: グループの識別子を指定します。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

GroupMemberProperties

PropertyData タイプの構造体の配列を返します。詳しくは、114 ページの " インストールグループメンバプロパティのレイアウト " と 114 ページの " グループメンバプロパティのインストール表 " をご覧ください。

5.9 デプロイメント管理

原則として、Job API にはデプロイメントのための手順がいくつかあります。

– クローニング

  クローニングとは、マスタシステムにインストールしてマスタシステムのイメージファイルを作成し（79 ページの「CreateImage」の項をご覧ください）、そしてこのイメージファイルを同じ構成パラメータでインストールするサーバグループへクローニングすることを意味します。

– 大規模なリモート インストール

  大規模なリモート インストールとは、Installation Manager で作成したコンフィグレーションファイルを使い、このファイルをサーバやサーバグループへ割り当て、そしてすべてのサーバに対するインストールを開始することを意味します。

5.9.1 Deploy

Deploy 関数を使ってグループにあるすべてのサーバ、指定した 1 つのサーバ、またはグループ内のサーバのリストのデプロイメントを管理します。一般的に認められているパラメータ（例 *DeplomentServer*）は通常の方法で指定する必要があります。また、その他のパラメータはパラメータ構造（*ParameterData*）の配列によって指定します。すべての既知パラメータのリストはグループの種類によって異なります。これに関しては次の節で説明しています。

Deploy 関数は新しいジョブを開始し、Job ID を返します。以降の *GetJobStatusDetails* の呼び出し入力としてこの Job ID を使用することができます。

単一サーバのデプロイメントを簡単にするために、Job API は次の 2 つのデプロイメントモードを識別します。

– シングルサーバモード
このモードは *GroupID* が 0 で、*ServerIdList* でサーバを 1 つだけ指定した場合、有効になります。したがって、所定の単一サーバだけがデプロイメントされるため、グループ管理は不要です。

– グループモード
グループモードを使うと、1 つの Deploy 関数を実行するだけで複数のサーバをデプロイメントすることができます。したがって、デプロイメントを実行する前に、グループ管理を使ってサーバが含まれる適切なグループを設定する必要があります。また、*ServerIdList* 経由で指定された所定サーバにある所定グループを縮小できます。

デプロイメントのプラットフォーム WinPE MDP でクローニング、またはインストールする場合、この関数には Installation Manager が必要です。

構文

```
Deploy GroupID=<int>
          ServerIdList=<ArrayOf_ServerIdentifier>
          Params=<ArrayOf_ParameterData>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

GroupID

既存グループの識別子を指定する、または単一サーバ選択モードで 0 を指定します。

ServerIdList

選択したグループ外の一時サブグループを定義するために、サーバのリストを指定します。サーバを指定しない場合、この関数はグループのすべてのサーバに対して適用されます。サーバを指定するには、*ServerIdentifier* タイプの構造体の配列を使用します。この構造体に関する定義は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

Params

ParameterData タイプの構造体の配列を指定します。構造体に関する詳細は 27 ページの "ArrayOf_ParameterData 構造体のレイアウト " をご覧ください。また、使用可能なパラメータについては 120 ページの " クローニングに使用できるパラメータ " と 127 ページの " インストールに使用できるパラメータ " をご覧ください。

DeploymentServer

デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウント名を指定します。

DeploymentPassword

デプロイメントサーバ上の指定したユーザアカウントに対するパスワードを指定します。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

ShutdownType が 3 で IPMI を搭載していないサーバを指定した場合、シャットダウンは手動で行う必要があるため、この関数は正常に実行されたと返します。

クローニングに使用できるパラメータ

デプロイメント方式「クローニング」を実行するためにパラメータの配列で指定できるパラメータを示します。

ImagePath

イメージ名を含む完全修飾パスを指定します。
このパラメータは単一サーバ選択モードでは必須です。

PreparationMode

準備モードを指定します。
1 (PREPARATIONMODE_BLADEOPTIMIZED):
2 (PREPARATIONMODE_ALLPRIMERGY):
3 (PREPARATIONMODE_GENERICBOOTIMAGE):
4 (PREPARATIONMODE_MANUAL):
このパラメータは必須です。
1 （*PREPARATIONMODE_BLADEOPTIMIZED*）は廃止になりました。使用できるのは legacy blade BX300 のみです。その他のブレードすべてに対しては、2（*PREPARATIONMODE_ALLPRIMERGY*）を使用してください。この場合、デプロイメントのプラットフォーム *WinPE MDP* を使用します（使用しない場合は、デプロイメント構成で別のモードを指定します）。

BootImage

選択したブートイメージの格納場所の完全修飾パスを指定します。
このパラメータは *PreparationMode* が 3（*PREPARATIONMODE_GENERICBOOTIMAGE*）の場合のみ、使用できます。またこのパラメータは必須です。

BootImageType

ブートイメージの種類を指定します。
0（BOOTIMAGETYPE_BOOTSTRAPIMAGE）: 複数の部分から成るブートイメージです。1 番目の部分は PXE ブートです。残りの部分は tftp ダウンロードで要求します。
1（BOOTIMAGETYPE_DOSIMAGE）: ブートイメージは「仮想」フロッピーイメージです（1.4MB ファイル）。
このパラメータは *PreparationMode* が 3（*PREPARATIONMODE_GENERICBOOTIMAGE*）の場合にのみ使用できます。
デフォルト値：BOOTSTRAPIMAGE

ControllerVendor

コントローラのベンダを指定します。
LSI/Mylex Raid Controller
Adaptec Raid Controller
Promise Raid Controller
Any Raid Controller（デプロイメントのプラットフォームが *WinPE MDP* の場合のみ）
Fujitsu Raid Controller（デプロイメントのプラットフォームが *WinPE MDP* の場合のみ）
ControllerVendor/ControllerFamily/ControllerName/RaidType に対して使用可能な値は Installation Manager（ファイル *RaidDef.xml* をご覧ください）により定義されます。
ControllerVendor だけを指定して、*ControllerFamily* と *ControllerName* を省略することは可能です。同じベンダでもファミリや名前が異なる複数のコントローラがご使用のコンピュータにあり、どれを使用するか明確に指定したい時は、この 2 つのパラメータのうち一方または両方を指定します。
このパラメータは、*PreparationMode* が 2（*PREPARATIONMODE_ALLPRIMERGY*）の場合のみ使用でき、また必須です。

ControllerFamily

コントローラファミリを指定します。
このパラメータは、*PreparationMode* が 2（*PREPARATIONMODE_ALLPRIMERGY*）の場合のみ使用でき、また必須です。
デフォルト値：""

ControllerName

コントローラの名前を指定します。
このパラメータは、*PreparationMode* が 2（*PREPARATIONMODE_ALLPRIMERGY*）の場合のみ使用でき、また必須です。
デフォルト値："

ControllerNumber

システムで定義されている通りに（所定のベンダ／ファミリ／名前の選択に応じて）論理コントローラを指定します。このパラメータは *PreparationMode* が 2（*REPARATIONMODE_ALLPRIMERGY*）の場合のみ使用できます。
デフォルト値：0

RaidType

Raid 構成を指定します。このパラメータはオプションです。
PreparationMode が 1（*PREPARATIONMODE_BLADEOPTIMIZED*）の場合：
-1 (RaidType unchanged): Raid 構成を行いません。
0 (RaidType_0)
1 (RaidType_1)
2000 (RaidType_2_Raid0_Arrays)
デフォルト値：0 (RaidType_0)

PreparationMode が 2（*PREPARATIONMODE_ALLPRIMERGY*）の場合：
使用可能な値は Installation Manager（ファイル *RaidDef.xml* をご覧ください）によって定義されます。
パラメータを指定しない場合は、*Automatic Mode* になります。このパラメータを指定した場合、*Manual Mode* になり、オプションの *DiskNumber* と *HotSpare* が追加で指定できます。

RaidInit

指定した Raid 構成の初期化を指定します。
0 (RAIDINIT_OFF)
1 (RAIDINIT_ON)
デフォルト値：0 (RAIDINIT_OFF)
このパラメータは、*PreparationMode* が 1（*PREPARATIONMODE_BLADEOPTIMIZED*）の場合のみ使用できます。

DiskNumber

ディスクの数を指定します（0 はすべてのディスクを意味します)。
デフォルト値：0
このパラメータは、*PreparationMode* が 2（*PREPARATIONMODE_ALLPRIMERGY*）で、パラメータ *RaidType* が設定されている場合のみ使用できます。

HotSpare

故障したハードディスクの交換に使用するスタンバイハードディスクを指定します。
デフォルト値：0
このパラメータは、*PreparationMode* が 2（*PREPARATIONMODE_ALLPRIMERGY*）で、パラメータ *RaidType* が設定されている場合のみ使用できます。

ShutdownType

サーバが動作している場合の応答を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。ただし、サーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI):
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
3 (SHUTDOWNTYPE_FORCE): ブレードタイプのサーバに対しては、マネージメントブレードによって強制的に電源を切断します。その他のタイプのサーバに対しては、ACPI で電源を切断します。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownCommunity

対象サーバ上で動作している ServerView SNMP エージェントに対して発行するシャットダウン要求の SNMP コミュニティを指定します。
デフォルト値："

ShutdownUsername

ServerView SNMP エージェントシャットダウンに対するユーザ名を指定します。
ShutdownType が 2（*graceful*）の場合、このパラメータは必須です。
デフォルト値："

ShutdownPassword

前に指定したユーザに対するパスワードを指定します。
ShutdownType が 2（*graceful*）の場合、このパラメータは必須です。
デフォルト値："

FinalPowerStatus

クローニング後の電源状態を指定します。
0 (FINALPOWERSTATUS_OFF)
1 (FINALPOWERSTATUS_ON)
デフォルト値：0 (FINALPOWERSTATUS_OFF)

CustomScriptCmdline

クローニング後に実行する、カスタマー（お客様）が作成したスクリプトを指定します。
対象システムが Windows の場合、パス名はスクリプトのネットワークパス名を UNC 表記法で指定します。スクリプトが格納されているネットワーク共有を使用しなければならないことを意味します。このスクリプトに対し、オプションのパラメータを指定することもできます。
対象システムが Linux の場合、次の書式でパス名を指定します。
\\<server>\<path>\script_name [parameters] または
//<server>/<path>/script_name [parameters]
デフォルト値：""

CustomScriptUsername

対象システムのカスタマースクリプトにアクセスするために使用するアカウント情報を指定します。対象システムが Windows の場合、ユーザアカウントには共有と指定サブディレクトリに対する読み込みおよび実行許可が必要です。対象システムが Linux の場合、このアカウントを使って FTP サーバへアクセスします。カスタマースクリプトを指定した場合、このパラメータは必須です。
デフォルト値：""

CustomScriptPassword

指定したカスタムスクリプトのユーザアカウントに対するパスワードを指定します。カスタマースクリプトを指定した場合、このパラメータは必須です。
デフォルト値：""

WindowsDomain

クローニング後に参加する Windows ドメインを指定します。
デフォルト値：""

WindowsDomainJoinUsername

ドメインに参加する際に使用するユーザ名を指定します。このユーザアカウントには管理者権限が必要です。
デフォルト値：""

WindowsDomainJoinPassword

指定した *WindowsDomainJoinUsername* アカウントに対するパスワードを指定します。
デフォルト値："""

LogicalDisk

イメージをデプロイメントする論理ディスク番号を指定します。
デフォルト値：0

ExpandLastPartition

ディスクの空き領域を使用するために、イメージの最後のパーティションの拡大を有効または無効にします。
0 (EXPANDLASTPARTITION_OFF): 最後のパーティションを拡張しません。未使用のスペースはその他のパーティション用に使用可能な状態のままにします。
1 (EXPANDLASTPARTITION_ON): クローニングされるシステムの使用可能ディスク容量まで、最後のパーティションを拡大します。
デフォルト値：0 (EXPANDLASTPARTITION_OFF)

Unicast

マルチキャストまたはユニキャストクローニングの実行を指定します。
0 (MULTICAST)
1 (UNICAST)
デフォルト値：0 (MULTICAST)

GUIDCheck

グローバルユーザ ID の確認を有効または無効にします。
0 (GUIDCHECK_OFF)
1 (GUIDCHECK_ON)
デフォルト値：0 (GUIDCHECK_OFF)

IgnoreModelCompatibility

クローニングイメージが所定の富士通サーバと互換性があるかどうかを確認する、互換性チェックを有効または無効にします。
0 (IGNORE_OFF) 互換性チェックを実行します。
1 (IGNORE_ON) 互換性チェックを無視します。
デフォルト値：0 (IGNORE_OFF)

ForceSnapshotCloning

スナップショット以外のイメージに対して通常行われるクローニング済みシステムの後処理（ホスト名、IP アドレスなどの設定）を行わずに、スナップショット以外のイメージをディスクへクローニングするというオプションを有効または無効にします。
0 (FORCESNAPSHOT_OFF)
1 (FORCESNAPSHOT_ON)
デフォルト値：0 (FORCESNAPSHOT_OFF)

SetEtcHostname

Linux RedHat をクローニングする時のみに使用します。RedHat のバージョンによっては、*/etc/hosts* にある localhost の行にホスト名を追加するものもあります。このパラメータを使って、クローニング後、localhost の行を強制的に変更できます。
0 （またはパラメータを指定しない場合）: デフォルトの対応です。ほとんどの場合適応されます。
1: クローニングされたシステムのホスト名を localhost の行へ追加します。
2: localhost の行からホスト名を削除します。
デフォルト値：0

MakKey

Windows2008 でのみ使用可能です。
このパラメータには Multiple Activation Key が含まれており、クローニング後、Windows 2008 対象を自動的に有効化します。
デフォルト値：""

KmsServer

Windows2008 でのみ使用可能です。
このパラメータには「Key Management Service」サーバの IP アドレスまたはホスト名が含まれており、クローニング後、Windows2008 対象を自動的に有効化します。
デフォルト値：""

KmsPort

Windows2008 でのみ使用可能です。
Windows2008 でのみ使用可能です。「Key Management Service」のポート番号です。パラメータ *KmsServer* が設定されている場合このパラメータを使用します。
デフォルト値：""

ProxyServer

Windows2008 でのみ使用可能です。
Windows2008 でのみ使用可能です。パラメータ *MakKey* または *KmsServer* のどちらかを設定した場合、オプションでこのパラメータを設定できます。プロキシサーバの IP アドレスまたはホスト名を指定できます。
デフォルト値："""

ProxyPort

Windows2008 でのみ使用可能です。
プロキシサーバのポート番号です。パラメータ *ProxyServer* を設定した場合は、このパラメータを指定してください。
デフォルト値："""

インストールに使用できるパラメータ

デプロイメント方式「インストール」を実行するためにパラメータの配列で指定できるパラメータを示します。

ConfigPath

コンフィグレーションファイル名を含む完全修飾パスを指定します。
シングルサーバ選択モードでは、このパラメータは必須です。

ShutdownType

サーバが動作中の場合の応答を指定します。
0 (SHUTDOWNTYPE_NO): シャットダウンしません。しかしサーバの電源が切断されていない場合はエラーを返します。
1 (SHUTDOWNTYPE_ACPI):
2 (SHUTDOWNTYPE_GRACEFUL): ServerView エージェントを使ってシャットダウンします。
3 (SHUTDOWNTYPE_FORCE): ブレードタイプのサーバに対しては、マネージメントブレードによって強制的に電源を切断します。その他のタイプのサーバに対しては、ACPI で電源を切断します。
デフォルト値：0 (SHUTDOWNTYPE_NO)

ShutdownCommunity

対象サーバで稼動している ServerView SNMP エージェントに対するシャットダウン要求の SNMP コミュニティを指定します。
デフォルト値："""

ShutdownUsername

ServerView SNMP エージェントシャットダウン用のユーザ名を指定します。
ShutdownType が 2（*graceful*）の場合、このパラメータは必須です。
デフォルト値："

ShutdownPassword

前に示したユーザに対するパスワードを指定します。
ShutdownType が 2（*graceful*）の場合、このパラメータは必須です。
デフォルト値："

FinalPowerStatus

インストール後のサーバの電源状態を指定します。
0 (FINALPOWERSTATUS_OFF): サーバの電源は切断されます。
1 (FINALPOWERSTATUS_ON): インストールされている OS が起動されます。
このパラメータを使って、インストール後、サーバの OS を起動する、または サーバの電源を切断するかを選択します。
デフォルト値：0 (FINALPOWERSTATUS_OFF)

SeStContentsPath

Installation Manager コンテンツツリーの完全修飾パスを UNC 形式で指定します。
このパラメータを指定しない場合、Installation Manager コンテンツツリーのデフォルト値が使用されます。

SeStContentsUsername

Installation Manager データへアクセスするためのユーザ名を指定します。このパラメータを指定しない場合、リモートアクセス用のユーザが使用されます。

SeStContentsPassword

指定したユーザ（または *SeSetContentsUsername* が設定されていない場合はデフォルトユーザ）に対するパスワードを指定します。
このパラメータは必須です。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 になります。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

JobID

詳細を取得するために *GetJobStatusDetails* に対して使用する、ユニークな ID です。

5.10 ジョブ管理

Job API は、ジョブを要求とみなします。この要求は、デプロイメントエンジンに送信され、バックグラウンドで実行されます。ジョブはそれぞれに対して作成された JobID と関連付けられます。

– イメージ作成 (*CreateImage*)

– イメージ割り当て (*Deploy*)

– システムインストール (*Deploy*)

– システム情報の取得 (*StartSystemInfoRetrieval*)

さらに、Job API はジョブスケジューリングもサポートします。汎用 Job API 関数 *Deploy* と *CreateImage* により定義されるジョブは、即時、毎日、毎週、毎月、またはスケジュールジョブ管理プロパティを使って設定した予定での実行をスケジューリングします。スケジューリング管理プロパティについての詳細は、「6 ジョブスケジューリング」の章（145 ページ）をご覧ください。

Job API はジョブの種類を 2 つに区別します。

– スケジュールジョブ
現在のバージョンの Job API でのスケジュールジョブは、関数 *Deploy* と *CreateImage*、およびスケジュールジョブ管理プロパティを追加で指示することにより定義されます（これらのジョブは、指定したスケジューリング管理プロパティに応じてデプロイメントエンジンへ送信されます）。

– スケジュールされていないジョブ
スケジュールされていないジョブとは、デプロイメントエンジンに対して要求され、バックグラウンドで直接開始されたジョブのことです。

5.10.1 GetJobStatusDetails

Job API により実行されたジョブの詳細をジョブ開始以降に取得するには、次の関数を使用します。*GetJobStatusDetails* は Job ID または Job ID リストで識別された現在のジョブ状態と詳細を返します。1 つのジョブの実行が完了した時、またはタイムアウト値が経過した時に返します。API 関数は非同期性です。同期して進めたい場合は、*Timeout* を *INFINITE* に設定し *GetJobStatusDetails* を発行します（ただし、ほとんどの *SoapClinents* と *SoapServers* は、自身でタイムアウトを設定する必要があることにご注意ください）。*GetJobStatusDetails* の複数の呼び出しは並行して発行できます。

構文

```
GetJobStatusDetails JobIDList=<string>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          Timeout=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

JobIDList

Deploy、*CreateImage* など以前呼び出した Job ID に関するコンマ区切りのリストを指定します。

DeploymentServer

デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウント名を指定します。このパラメータは必須です。

DeploymentPassword

デプロイメントサーバで指定したユーザアカウントに対するパスワードを指定します。このパラメータは必須です。

Timeout

ジョブが完了していない場合の待機時間を秒単位で指定します。
0: すぐに返します。
1（INFINITE）: ジョブが完了するまで待ちます。
デフォルト値：0

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

複数のサーバで使用するジョブでは、*GetJobStatusDetails* は関連するサーバの結果を *Success* または *Error* で返します。*Error* である場合、1 つ以上のサーバのデプロイメントに失敗したことを示しています。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

JobStatusDetailsList

JobStatusDetailsData タイプの構造体の配列を返します。

戻りリストのレイアウト

戻りリストには、指定した Job ID のジョブ情報がすべて含まれています。またレイアウトは次のとおりです。

```
JobID=<int> JobType=<int>
StatusDetails=<ArrayOf_DetailData>
```

JobID

Job ID を表示します。

JobType

Job の種類を表示します。
0 (JOBTYPE_STARTSYSTEMINFORETRIEVAL)
8 (JOBTYPE_CREATEIMAGE)
9 (JOBTYPE_DEPLOY)

StatusDetails

DetailData タイプの要素を使った配列で指定した Job ID の状態に関する詳細をすべて表示します。詳細は、133 ページの "DetailData 構造体のレイアウト " をご覧ください。

DetailData 構造体のレイアウト

```
Name=<string> Value=<string>
```

Name

状態詳細の名前を表示します。

Value

状態詳細の値を表示します。

5.10.2 GetServerJobStatus

関数 *GetServerJobStatus* はサーバの所定 MAC アドレスリストに基づいて、現在のジョブ状態（サーバ上の最新のジョブ）を返します。API 関数は非同期性です。同期して進めたい場合は、*Timeout* を *INFINITE* に設定し *GetServerJobStatus* を発行します（ただし、ほとんどの *SoapClients* と *SoapServers* はタイムアウト値をご自身で設定する必要があるのでご注意ください）。*GetServerJobStatus* への複数の呼び出しは並行して発行できます。

構文

```
GetServerJobStatus ServerIdList=<ArrayOf_ServerIdentifier>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          Timeout=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ServerIdList

Deploy、*CreateImage* などのジョブ呼び出しがすでに開始された 1 つ以上のサーバを認識すつための *ServerIdentifier* 構造体の配列を指定します。構造体に関する詳細は、26 ページの "ServerIdentifier 構造体のレイアウト " をご覧ください。

DeploymentServer

デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウント名を指定します。このパラメータは必須です。

DeploymentPassword

デプロイメントサーバで指定したユーザアカウントに対するパスワードを指定します。このパラメータは必須です。

Timeout

ジョブが完了していない場合の待機時間を秒単位で指定します。
0: すぐ返します。
-1 (INFINITE): ジョブが完了するまで待ちます。
デフォルト値：0

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

ServerJobStatusList

ServerIdentifier タイプの要素を使った配列を返します。

Layout of the returned list

戻りリストには指定したサーバ ID のジョブ状態情報が含まれています。レイアウトは次のとおりです。

JobID

指定したサーバで最後に実行されたジョブのジョブ ID を表示します。

Status

現在のジョブ状態を表示します。
"Initializing"
"Waiting"
"Running"
"'Cancelling"
"Success"
"Error"
"Cancelled"

Type

ジョブの種類を表示します。
0 (JOBTYPE_STARTSYSTEMINFORETRIEVAL)
8 (JOBTYPE_CREATEIMAGE)
9 (JOBTYPE_DEPLOY)

DetailedStatusCode

より詳しい状態コード番号（現在は常に 0）を表示します。

DetailedStatus

より詳しい状態（ジョブがすでに終了している場合のエラーメッセージの詳細やジョブ進行状況の記述）を表示します。

現在まで、指定したサーバに対してジョブを実行していない場合、返される JobID は 0 です。

5.10.3 CancelJob

状況によっては、キャンセルできるジョブもあります。そのようなジョブをキャンセルするには、ユーザが Job API 関数 *Cancel Job* を使い、実行中のジョブをキャンセルします。

構文

```
CancelJob JobID=<int>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

JobID

Deploy、*CreateImage* などの関数を前回呼び出した際に取得した JobID を指定します。

DeploymentServer

デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウント名を指定します。このパラメータは必須です。

DeploymentPassword

デプロイメントサーバで指定したユーザアカウントに対するパスワードを指定します。このパラメータは必須です。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

5.10.4 GetScheduledJobProperties

関数 *GetScheduledJobProperties* は、*ScheduledJobID* により識別されすでに定義されたスケジュールジョブのスケジュールジョブ管理プロパティを取得します。

構文

```
GetScheduledJobProperties ScheduledJobID=<int>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ScheduledJobID
: スケジュールジョブを定義する *Deploy*、*CreateImage* などの関数を前回呼び出した際に取得した *ScheduledJobID* を指定します。

DeploymentServer
: デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername
: 指定したデプロイメントサーバのユーザアカウント名を指定します。このパラメータは必須です。

DeploymentPassword
: デプロイメントサーバで指定したユーザアカウントに対するパスワードを指定します。このパラメータは必須です。

RDServer
: ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey
: 特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

Properties

スケジュールジョブ管理プロパティのリストを返します。

5.10.5 GetScheduledJobLogList

関数 *GetScheduledJobLogList* を使って所定のスケジュールジョブにより開始されたジョブに関する情報の詳細を返します。

構文

```
GetScheduledJobLogList ScheduledJobID=<int>
          DeploymentServer=<string>
          DeploymentUsername=<string>
          DeploymentPassword=<string>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ScheduledJobID

スケジュールジョブを定義した *Deploy*、*CreateImage* などの関数を前回呼び出した際に取得した *ScheduledJobID* を指定します。

DeploymentServer

デプロイメントサーバのホスト名、または、1 つ目の LAN ポートの IP アドレスを指定します。

DeploymentUsername

指定したデプロイメントサーバのユーザアカウント名を指定します。このパラメータは必須です。

DeploymentPassword

デプロイメントサーバで指定したユーザアカウントに対するパスワードを指定します。このパラメータは必須です。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："""

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

LogList

指定したスケジュールジョブのログすべての配列を返します。ログには、この定義されたスケジュールジョブにより開始されたジョブの詳細が含まれています。

戻りログリストのレイアウト

ログの戻りリストには開始されたジョブの詳細が含まれています。

```
JobID=<int> Details=<ArrayOf_DetailData>
```

JobID

スケジュールジョブにより開始されたジョブの JobID を表示します。

Details

JobID により指示されたジョブの詳細を表示します。構造体に関する詳細は、133 ページの "DetailData 構造体のレイアウト " をご覧ください。

5.10.6 DeleteScheduledJob

定義されたスケジュールジョブを削除するには、関数 *DeleteScheduledJob* を使用します。

構文

```
DeleteScheduledJob ScheduledJobID=<int>
          RDServer=<string>
          UserKey=<string>
```

パラメータ

ScheduledJobID

スケジュールジョブを定義した *Deploy*、*CreateImage* などの関数を前回呼び出した際に取得した *ScheduledJobID* を指定します。

RDServer

ServerView Deployment Manager サービスが稼動しているサーバのホスト名または IP アドレスを指定します。このパラメータを指定しない場合、ローカルのマシンが使用されます。
デフォルト値："

UserKey

特定のユーザアカウント専用ユーザキーを指定します。

戻り値

ErrorCode

最後に実行した際の診断番号です。正常に実行された場合、*ErrorCode* は 0 に設定されます。

ErrorString

ErrorCode のテキスト記述です。正常に実行された場合、*ErrorString* は空白です。

6 ジョブスケジューリング

Job API はジョブスケジューリング機能をサポートします。一般的な Job API 関数 *Deploy* と *CreateImage* は、即時、毎日、毎週、毎月、またはスケジュールジョブ管理プロパティを使って設定されたタイミングで実行するように、スケジューリングできます。

次の表で、パラメータ（名前 / 値のペア）として Job API 関数で使用できるスケジュールジョブ管理プロパティについて説明します。これらのプロパティによって、1 回のみの実行、またはジョブを繰り返し実行する際の日付や時間を設定します。

定義済みのスケジュールジョブのプロパティは、適切な Job API 関数（*Deploy* または *CreateImage*）を繰り返し実行する際に、*ScheduledJobID* を追加で指示する、またはスケジュールジョブ管理プロパティを全て選択することで変更できます。次回このスケジュールジョブを実行すると、修正されたスケジュールジョブで開始されます。さらに、すでに定義されたスケジュールジョブを *DeleteScheduledJob* 関数で削除し、適切な関数を実行する、またはスケジュールジョブ管理プロパティで有効な組み合わせを指定する方法もあります。

原則として、スケジュールジョブは定義された Job API 関数により指定された専用の *UserKey* に設定されています。すなわち、適切な *UserKey* の指定を前提に、既存のスケジュールジョブを修正または削除する点に注意してください。

パラメータ名	パラメータの説明と値
Scheduled	スケジュールジョブのオプションを有効または無効にします。スケジューリングを定義するには、追加でスケジュールジョブ管理プロパティを指定する、または管理プロパティのデフォルト値を使用します。 "0": すぐにジョブを実行する。 "1": スケジュールジョブを作成する。 デフォルト値："0"
ScheduledJobName	新しいスケジュールジョブの名前を指定します。
ScheduledJobID	すでに定義されたスケジュールジョブのジョブ ID を指定します。このパラメータを使って、すでに定義されたスケジュールジョブを修正できます。

パラメータ名	パラメータの説明と値
Retries	スケジュールジョブが失敗した場合の再試行回数を指定します。 デフォルト値：2
RetriesInterval	スケジュールジョブの試行に失敗した場合、次の試行までの時間（分単位）を指定します。再試行回数は、指定したスケジュールジョブの試行回数を超えることはありません。 数値範囲：1 ～ 360 分 デフォルト値：15
ScheduledStartTimeWindow	時間ウィンドウの範囲（分単位）を指定します。時間ウィンドウの範囲はスケジュールジョブの開始時から始まり、xx 分後（xx は StartTimeWindow に対して指定された値です）に終了します。時間ウィンドウ内でスケジュールジョブが正常に開始されなかった場合、そのスケジュールジョブは中止されます。また、スケジュールジョブはこの時間ウィンドウ内でのみ再試行されます。したがって、ジョブ開始時間ウィンドウを経過すると、スケジュールジョブは開始されません。ジョブをデプロイメントサーバへ渡すことはできますが、すでに大量のジョブが動作中のため、渡されたジョブは待ち行列に入れられます。この状態のままジョブ開始時間ウィンドウが経過した場合、タスクも中止されます。ジョブ開始ウィンドウが経過しても実行中のジョブは中止されません。 数値範囲：10 ～ 600 分 デフォルト値：60
ScheduledInterval	スケジュールジョブの間隔を指定します。 "Once": ジョブは 1 度だけ実行されます。 "Daily": ジョブは毎日実行されます。 "Weekly": ジョブは毎週実行されます。 "Monthly": ジョブは毎月実行されます。
ScheduledTime	スケジュールジョブの開始時間（時間と分単位）を指定します。ScheduledTime のフォーマットは hh:mm です。 デフォルト値：

パラメータ名	パラメータの説明と値
ScheduledDate	スケジュールジョブの開始日を指定します。ScheduledDate のフォーマットは、mm/dd/yyyy です。 デフォルト値：current date
ScheduledDays	1 週間のうちの 1 日、またはコンマで区切られた曜日のリストを使って、スケジュールジョブの実行予定を指定します。毎週実行するように予定されているジョブ（*ScheduledInterval*= Weekly）に対しては、次のとおりに曜日を指定します。 "SUN" | "1": 日曜日 "MON" | "2": 月曜日 "TUE" | "3": 火曜日 "WED" | "4": 水曜日 "THU" | "5": 木曜日 "FRI" | "6": 金曜日 "SAT" | "7": 土曜日 毎日実行するように予定されているジョブに対しては「WEEKDAYS」と指定します。「WEEKDAYS」とは、月曜日から金曜日までの毎日です。 *ScheduleDailyInterval* > 1 の場合、ScheduledDays の指定はできません。 デフォルト値：current date
ScheduledMonths	1 つの月、またはコンマで区切られた月のリストを使って、スケジュールジョブの実行予定を指定します。次のとおりに月を指定します。 "JAN" | "1": 1 月 "FEB" | "2": 2 月 .. "NOV" | "11": 11 月 "DEC" | "12": 12 月 毎月実行を予定しているジョブ（*ScheduledInterval*=Monthly）に対してこのパラメータを指定します。*ScheduledMonthlyInterval* > 1 が選択されている場合、ScheduledMonths は追加指定できません。 デフォルト値：""

パラメータ名	パラメータの説明と値
ScheduledDayOfMonth	スケジュールジョブを実行する月内の 1 日を指定します。毎月実行を予定しているジョブに対してこのパラメータを指定します。ただし、指定できるのは、パラメータ *ScheduledDayOfMonth* または *ScheduledWeekdayOfMonth* のどちらかのみです。 デフォルト値：""
ScheduledWeekdayOfMonth	スケジュールジョブを実行する月内の週または曜日を指定します（第 1 月曜日、第 2 水曜日など）。毎月実行を予定しているジョブ（*ScheduledInterval=Monthly*）に対してこのパラメータを指定できます。ただし、指定できるパラメータは *ScheduledDayOfMonth* または *ScheduledWeekdayOfMonth* のどちらかのみです。 デフォルト値：""
ScheduledWeeklyInterval	スケジュールジョブを実行する間隔を週単位で指定します。間隔が 1 の場合、スケジュールジョブは毎週実行されます。間隔が 2 の場合、スケジュールジョブは隔週に実行されます。 数値範囲：1 ～ 52 デフォルト値：0
ScheduledMonthlyInterval	スケジュールジョブを実行する間隔を月単位で指定します。間隔が 1 の場合、スケジュールジョブは毎月実行されます。間隔が 2 の場合、スケジュールジョブは隔月に実行されます。 数値範囲：1 ～ 12 デフォルト値：0

7 API エラーコード

ErrorCode や *ErrorString* パラメータ内で、Job API の呼び出しに対する API エラーコードが返される場合があります。それぞれの API 関数コールの詳細が定義されている特定情報は、故障やその他の原因を特定する際に役立ちます。

エラーコードや様々なエラーの原因についての説明範囲は多岐に渡るため、ここではエラーコードの説明は記載していません。エラーや対処方法についての詳細を入手するには、Deployment Manager 開発チームにご連絡ください。

8 インストール

Job API は Deployment Manager CD-ROM 上でアドオンパッケージとして配布されます。CD-ROM を挿入すると、インストールは自動的にスタートします。

Job API を構成しているインストールパッケージは次の 2 つです。

– *Job API for IIS* パッケージ

– *Job API for Apache* パッケージ

これらのパッケージは、同時にまたは別々にインストールできます。Job API パッケージを使ってデプロイメントサーバへアクセスできるため、Deployment Manager として別のシステムにインストールできます。Job API パッケージを別々の場所にインストールする場合、両方のコンピュータは同じ Windows のネットワークドメイン内にある必要があります。さらに、インストール中に指定するユーザアカウントはこのネットワークドメインのメンバである必要があります。

Job API for Apache パッケージをインストールする場合は、Apache Web サーバが ServerView Operations Manager セットアップと一緒にインストールされた場合にのみ、JOB API の Apache Web サーバ統合を行なうことができます。Job API と ServerView Operations Manager は同一のシステムにインストールする必要があります。

8.1 システム要件

Job API をインストールできるシステムは次のとおりです。

– Windows 2003

– Windows 2008

追加要件：

– Web サーバ (IIS または ServerView Apache)

8.2 インストール

この項では、*Job API for IIS* パッケージのインストール例を説明します。

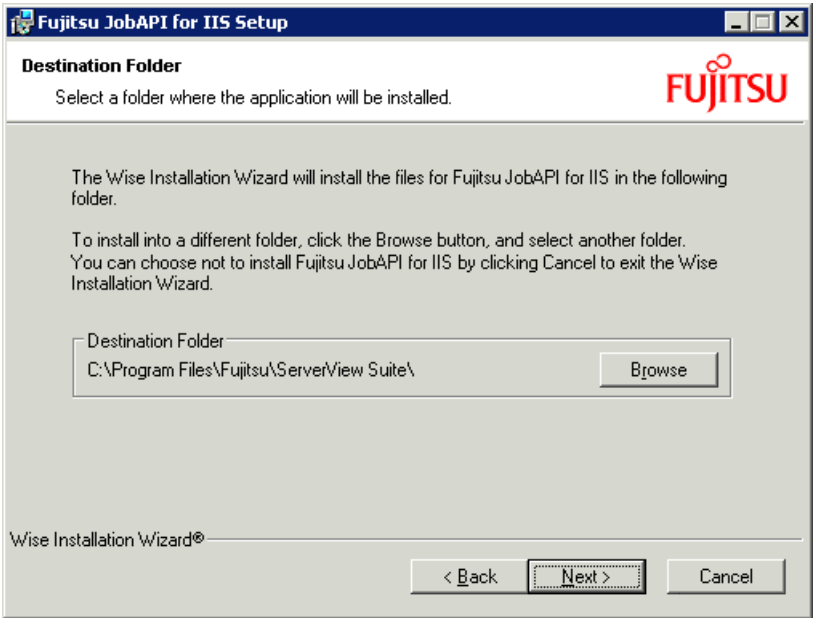
パッケージをインストールする場合は、次の手順に従ってください。

1. パッケージのコントロールボタンを有効にします。次の画面が表示されます。

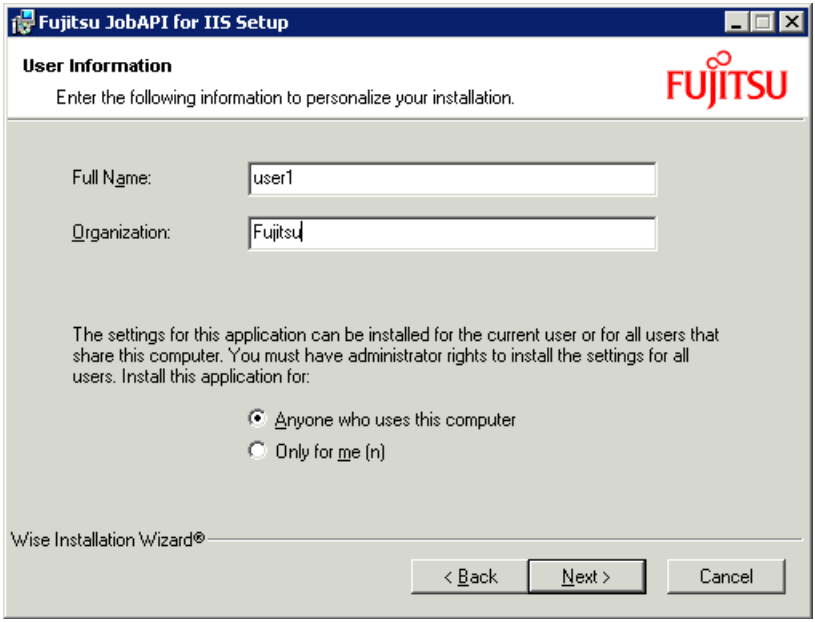
「Next」ボタンをクリックし、インストールを続行します。セットアッププログラムを終了したい場合は、「Cancel」ボタンをクリックします。

2. 適切なラジオボタンを選択してライセンス契約を承認し、「Next」ボタンをクリックします。

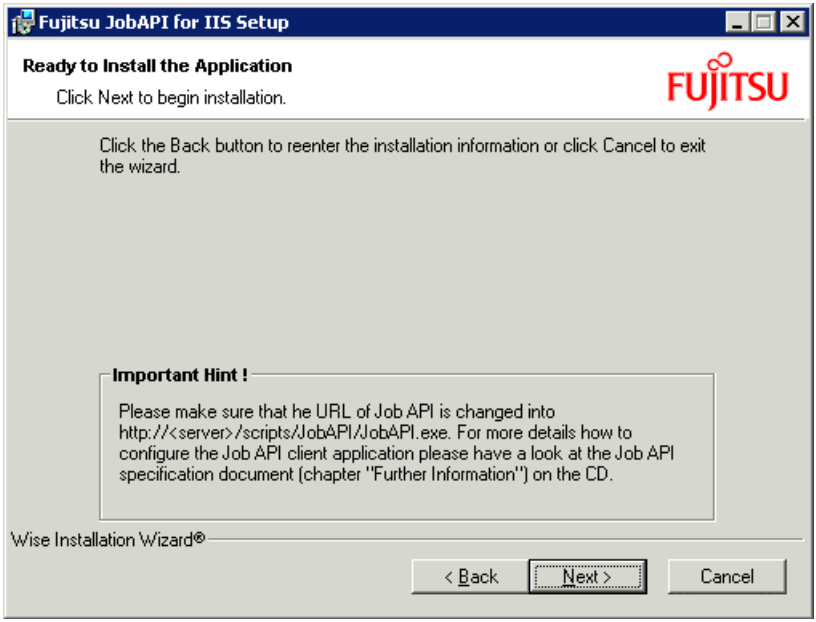
3. パッケージのインストール先フォルダを確認し、「Next」ボタンをクリックします。



4. 氏名と所属部署を入力します。Job API の設定を現在のユーザのみに適用するか、対象コンピュータを使うユーザ全員に適用するかを指定します。どちらかをラジオボタンで選択し、「Next」ボタンをクリックします。



5. インストールを始めるには、「Ready to install the Application」ウィンドウの「Next」ボタンをクリックします。
   インストールを中止したい場合は、「Cancel」ボタンをクリックします。


Fujitsu JobAPI for IIS Setup
Ready to Install the Application
Click Next to begin installation.
FUJITSU
Click the Back button to reenter the installation information or click Cancel to exit the wizard.
Important Hint !
Please make sure that he URL of Job API is changed into http://<server>/scripts/JobAPI/JobAPI.exe. For more details how to configure the Job API client application please have a look at the Job API specification document (chapter "Further Information") on the CD.
Wise Installation Wizard®
< Back
Next >
Cancel

6. Job API のインストールが正常に完了したら、「Finish」ボタンをクリックしインストールを終了します。

8.3 JobAPI と Internet Information Server (IIS)

8.3.1 Windows Server 2003 での JobAPI と Internet Information Server (IIS)

Job API を Windows Server 2003 にインストールし、Internet Information Server (IIS) を Web サーバとして使用する場合、CGI スクリプトが実行できるように IIS のデフォルト設定を変更します。変更は、次の方法に従って行います。

- 管理ツールから、インターネットインフォメーション サービス（IIS）マネージャを起動します。
- ローカルシステムの「Web サービス拡張」を選択します。
- 「すべての不明な CGI 拡張」を選択します。
- 「許可」ボタンをクリックします。

8.3.2 Windows Server 2008 での JobAPI と Internet Information Server (IIS)

Job API を Windows Server 2008 にインストールし、Internet Information Server (IIS) を Web サーバとして使用する場合、実行する CGI スクリプトに *JobAPI.exe* を追加してください。追加は、次の方法に従って行います。

- 「スタート」→「管理ツール」→「インターネットインフォメーション サービス（IIS）マネージャー」の順に選択します。
- 左側のウィンドウでご使用のサーバをクリックします。
- 中央のウィンドウで「ISAPI および CGI の制限」をダブルクリックします。
- 右側のウィンドウで「追加 ...」ボタンをクリックします。
- 「ISAPI または CGI の制限の追加」ウィンドウで次の値を入力します。I
  - ISAPI または CGI パス : *C:¥Inetpub¥scripts¥JobAPI¥JobAPI.exe*
  - フィルタ名 : JobAPI
  - 「拡張パスの実行を許可する」をチェックし実行します。
- 「OK」ボタンをクリックします。

8.4 インストールテスト

インストール後、Web ブラウザから Job API サービスをテストできます。

1. ブラウザのロケーションバーに次のいずれかを入力します。

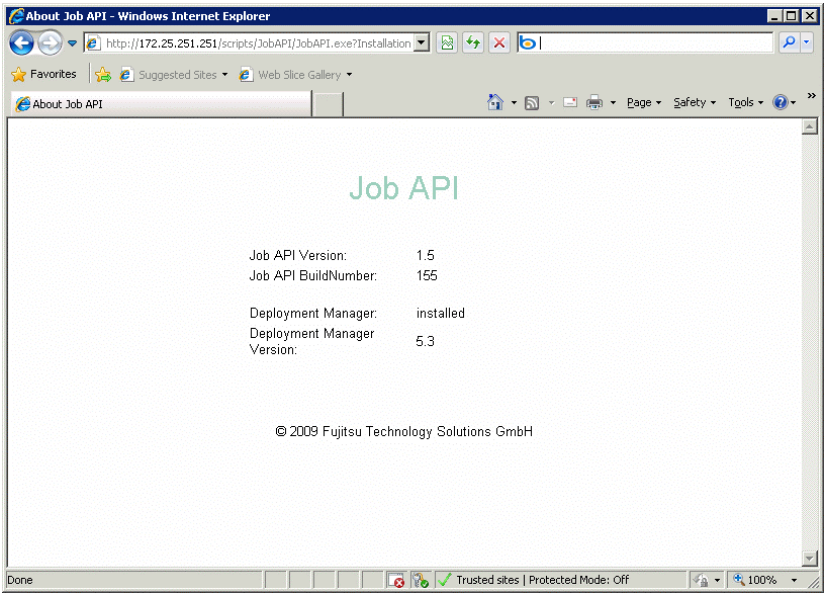
   http://<server>:<port>/scripts/JobAPI/JobAPI.exe?Installation

   または

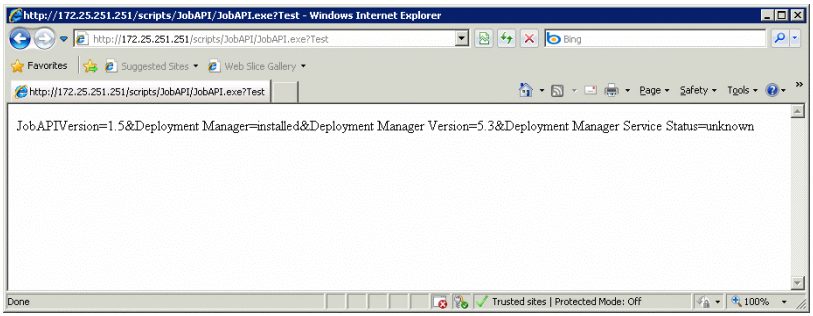
   http://<server>:<port>/scripts/JobAPI/JobAPI.exe?Test

   <server> は、Job API サービスをインストールしたサーバの IP アドレスまたはホスト名です。*<port>* は、Web サーバのポートです。Job API の URL に関する詳細は、161 ページの「URL」の項をご覧ください。

2. Job API サービスが正常にインストールされると、Job API CGI が起動し、次に示す Web ページまたは文字列が生成されます。



この Web ページでは、Job API のバージョン情報と同一サーバへの Deployment Manager のインストールに関する追加情報を確認できます。



表示された文字列には生成された Web ページと同じ情報が含まれますが、保存されるのは次の変数のみです。

JobAPIVersion	インストールされた Job API サービスのバージョンを表示します。
Deployment Manager	Job API のインストール先と同じサーバに Deployment Manager がインストールされているかを表示します。
Deploment Manager Version	インストールされている Deployment Manager のバージョンを表示します。
Deployment Manager Service Status	ServerView Deployment Manager サービスの状態を表示します（unknown ＝アクセスが拒否されたため確認できません）。

HTTP エラーが発生した場合は（例：404：見つかりません、400：無効な要求）、Job API がインストールされていない、または CGI 呼び出しが正しくないかのいずれかです。

9 その他の情報

次の章では、Job API クライアントアプリケーションと使用する Web サーバの設定のための重要な情報を説明します。

9.1 URL

Job API サーバは Internet Information Service（IIS）と Apache Web サーバにインストールできます。次の表は、Deployment Manager、Job API バージョン、構成と一緒にどの URL を使用する必要があるかについて説明しています。

Deployment Manager	Job API	開発用ツール	Web サーバ	URL
V5.30	V1.5	gSOAP V2.7.13	IIS	http://<server>/scripts/JobAPI/JobAPI.exe
		gSOAP V2.7.13	Apache (SV)	http://<server>:3169/scripts/JobAPI/JobAPI.exe

開発用ツールは、Job API サーバの生成用ツールに関する情報です。Job API クライアントの生成には、バージョン V2.6 ～ V.2.7.13 の gSOAP の使用を推奨します。

9.2 HTTP 認証

文字列 *soap.userid* と *soap.passwd* をそれぞれユーザ名とパスワードに対して設定することにより、Job API クライアント側で HTTP 認証ができるようになります。Job API クライアントが HTTP 認証なしで、または間違った認証情報で接続しようとすると、Web サーバはユーザ認証を求め、アクセス（HTTP401 エラー）を拒否する場合があります。

ServerView によってインストールされた Apache サーバのユーザおよびパスワードを設定するために使用する、Job API クライアントのコードフラグメント例を次で示しています。

```
struct soap soap;
soap init(&soap);
soap.userid = "svuser";
soap.passwd = "password";
```

クライアントの SOAP 要求には次の HTTP ヘッダが付きます。

```
POST /scripts/JobAPI/JobAPI.exe HTTP/1.1
Host: YYY:3169
User-Agent: gSOAP/2.7
Content-Type: text/xml; charset=utf-8
Content-Length: nnn
Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=
```

各呼び出し後、成功・失敗にかかわらず文字列がリセットされるため、クライアント認証が必要なそれぞれの呼び出しに対して、Job API クライアントは必ず文字列 *soap.userid* と *soap.passwd* を設定しなくてはなりません。

デフォルトインストールにより ServerView が Apache サーバを使用するよう設定されている場合、SSL サポートが有効になります。さらに、Job API サービスに対する HTTP 認証も有効になります。

HTTP 認証を無効にするためには、次に従って Apache サーバのコンフィグレーションファイル（*ssl.config*）を変更します。

1. SSL コンフィグレーションファイル（*C:¥Program Files¥Fujitsu¥ServerView Suite¥ServerView¥ServerView Services¥WebServer¥conf¥ssl.config*）を開きます。そして Job API クライアントがインストールされているサーバの IP アドレスをスクリプト認証の設定へ追加します。IP アドレス、または IP アドレスの範囲を入力できます。例えば、*122.138* は 呼び出されたサーバ *122.138.200.151* の IP アドレス範囲です。

   ssl.config file - 例

```
# scripts
   Directory    "C:/PROGRA~1/FUJITS~1/SERVER~1/SERVER~1/scripts">
      Options FollowSymLinks
      AllowOverride None

      Order deny,allow
      Deny from all
      Allow from 127.0.0.1
      AuthType Basic
      AuthUserFile "C:/PROGRA~1/FUJITS~1/SERVER~1/SERVER~1/WebServer/bin/passwd"
      AuthName "C:/PROGRA~1/FUJITS~1/SERVER~1/SERVER~1/WebServer/cgi-bin"
      Require valid-user
      Satisfy any
   </Directory>
```

   現在の SSL では、ローカルサーバからの Job API クライアントの要求だけを設定できます。サーバ *122.138.200.151* からのアクセスを許可するには、この IP アドレスを使って *Allow* ステートメントを拡張します。

   ```
   Allow from 127.0.0.1 122.138.200.151
   ```

   完全な IP アドレスの変更を許可するには、次を使って拡張します。

   ```
   Allow from 127.0.0.1 122.138.200
   ```

2. Apache Web サーバを再起動します。

9.3 HTTPS / SSL

「3 セキュリティ」の章（13 ページ）で説明されているとおり、Job API は CGI アプリケーションとして実行されます。Job API は Job API サービス CGI が稼動している Web サーバにより暗号化 / 複合化された標準的な I/O を使用します。したがって、Web サーバと Job API クライアントアプリケーションに対して HTTP/SSL サポートを設定する必要があります。

Job API クライアントが確実に HTTP/SSL を利用できるようにするには、OpenSSL ライブラリ（*http://www.openssl.org*）をご使用のプラットフォームにインストールしてください。
インストール後、オプション *-DWITH_OPENSSL* を使って Job API クライアントアプリケーションのソースをすべてコンパイルしてください。または、*soapdefs.h* に次の行を追加し、*-DWITH_SOAPDEFS_H* を使ってコンパイルしてください。

```
#define WITH_OPENSSL;
```

Job API サービスへの関数呼び出しのエンドポイント URL において暗号化された転送を行うために（Web サーバが HTTPS をサポートしている場合）、Job API クライアントアプリケーションは、単に *http:* の代わりとして接頭辞 *https:* を使用します。さらに、Web サーバは HTTPS プロトコル用に HTTP とは別のポートを使用します。デフォルトでは、IIS7.0 は HTTPS プロトコルに対してポート 443 で待機し、Apache2.0（ServerView）はポート 3170 で待機します。

さらに、クライアント側のキーファイルとキーファイルのパスワードを指定してください。指定方法については、OpenSSL のマニュアルをご覧ください。*http://www.openssl.org*.

```
if (soap_ssl_client_context(&soap,
    SOAP_SSL_DEFAULT,
    "client.pem",       // key file: required only when client must authenticate
                            to server
                        // (see SSL docs on how to obtain this file)
    "password",         // password to read the key file
    NULL,               // optional cacert file to store trusted certificates
    NULL,               // optional capath to direcoty with trusted certificates
    NULL                // if randfile!=NULL: use a file with random data to seed
                            randomness
    ))
 {
   soap_print_fault(&soap, stderr);
   exit(1);
 }
soap_call_ns_jobapifunction(&soap,
 "https://server:3169/scripts/JobAPI/JobAPI.exe", "", ...);
```

デフォルトでは、サーバ認証は有効になっています。テスト目的で、サーバ認証を無効にするには、次を使用します。

```
if (soap_ssl_client_context(&soap,
                            SOAP_SSL_NO_AUTHENTICATION,
                            NULL,
                            NULL,
                            NULL,
                            NULL,
                            NULL,
                            ))
     {
          soap_print_fault(&soap, stderr);
          exit(1);
     }
```

これもサーバがクライアントに認証（キーファイルがない）を要求しないことを前提にしています。アプリケーションでは、接続の切断（SIGPIPE）を検出するようにシグナルハンドラを設定してください。

```
signal(SIGPIPE, sigpipe_handle);
```

例：

```
void sigpipe_handle(int x) {…}
```

WITH_OPENSSL マクロを一貫して定義し、*stdsoap2.cpp*、*soapC.cpp*、*soapClient.cpp* などのソースと *stdsoap2.h* または *soapH.h* を含むすべてのアプリケーションソースをコンパイルすることが重要です。マクロの使用が一貫していないと、アプリケーションは、gSOAP 環境の宣言とアクセスの不一致によりクラッシュします。

9.4 SOAP タイムアウト

ソケットの接続、承認、送信、受信のタイムアウト値を設定すると、通信タイムアウトを管理できます。*soap.connect_timeout*、*soap.accept_timeout*、*soap.send_timeout*、および *soap.recv_timeout* 属性は、ユーザ定義された適切なソケット送信、受信、承認のタイムアウト値に設定できます。正数のタイムアウト値は秒単位で測定します。負数のタイムアウト値はマイクロ秒（10-6 秒）単位で測定します。

- *soap.connect_timeout* は soap call ns method 呼び出しのタイムアウト値を指定します。
- *soap.accept_timeout* は soap accept(&soap) 呼び出しのタイムアウト値を指定します。
- *soap.send_timeout* および *soap.recv_timeout* はブロックしないソケット I/O 処理のタイムアウト値を指定します。

例 :

```
struct soap soap;
soap init(&soap);
soap.send timeout = 10;
soap.recv timeout = 10;
```

送受信操作をソケット上で開始した後、10 秒以内にデータが一切送受信されなかった場合、タイムアウトになります。さらに、ゼロ値はタイムアウトを無効にします。
例 :

```
soap.send timeout = 0;
soap.recv timeout = 0;
```

送受信操作中にタイムアウトが発生した場合、SOAP_EOF 例外が出力されます（「end of file」、または「no input」）。

9.5 CGI タイムアウト

CGI タイムアウトは、Job API クライアントの同期要求が Job API サーバからの応答を待つ最大秒数を指定します。CGI がタイムアウト時間内に応答しない場合、要求処理は Web サーバにより削除されます。タイムアウトの特性は、*CreateImage*、*Deploy*、*StartSystemInfoRetrieval* 関数を使って行われた非同期要求に影響を与えません。

さらに、同期要求で示されているタイムアウト値よりも CGI タイムアウトが常に大きいことを確認してください。CGI タイムアウトが小さいと、関数タイムアウトが終了する前に Web サーバにより要求が削除される可能性があります。

サーバ固有のデフォルトタイムアウト値と手順に従い、タイムアウト間隔を増やしてください。

9.5.1 Apache 構成

Apache Web サーバ内の時間間隔は、*httpd.conf* のタイムアウト命令を使って修正します。この命令は次の 3 つに対して Apache Web サーバが待機する時間を定義します。

1. GET 要求を受信するために要する総時間。
2. POST または PUT 要求での TCP パケット受信間の時間。
3. 応答内で TCP パケットを送信する際の ACK 間の時間。

Apache1.2 以前は、1200 をデフォルトとしてタイマーを使用していましたが、ほとんどの状況において、より必要な数値である 300 に下げられました。

次のサーバ固有な手順に従ってタイムアウト間隔を増やします。

1. *httpd.conf* ファイルを開きます。
2. 次の図で示すとおりタイムアウト値を修正します。
3. *httpd.conf* ファイルを保存します。

```
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15
```

タイムアウト時間の変更を有効にするには、Web サーバを再起動する必要があります。

9.5.2 IIS 構成

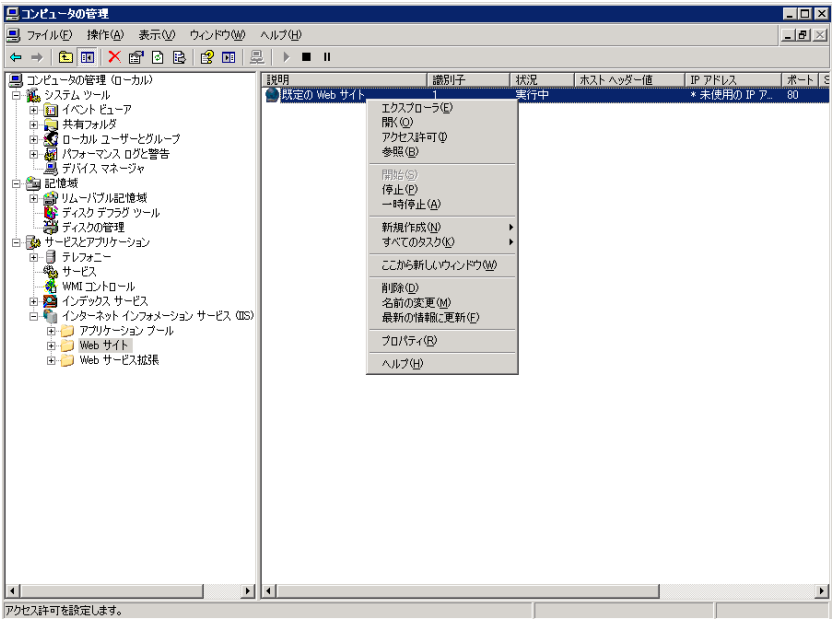
Microsoft Internet information service (IIS) の構成では、タイムアウトのデフォルト値は 900 秒に設定されています。

次のサーバ固有な手順に従ってタイムアウト間隔を増やします。

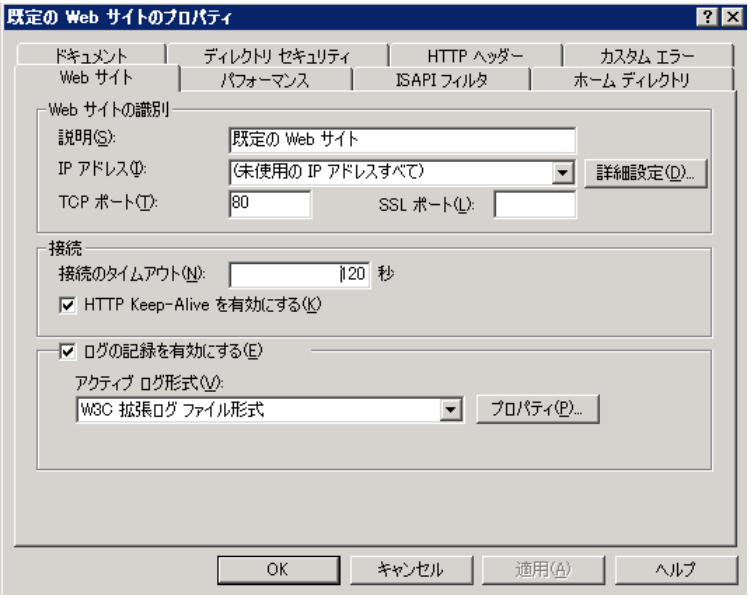
► インターネット インフォメーションサービス (IIS) マネージャを開きます。

次の図で示すような IIS ウィンドウが表示されます。

► 「Web サイト」フォルダを右クリックし、「プロパティ」をクリックします。



► 「接続」ボックス内で「接続のタイムアウト」タブへ進み、タイムアウト値を修正します。


既定の Web サイトのプロパティ
ドキュメント
ディレクトリ セキュリティ
HTTP ヘッダー
カスタム エラー
Web サイト
パフォーマンス
ISAPI フィルタ
ホーム ディレクトリ
Web サイトの識別
説明(S):
既定の Web サイト
IP アドレス(I):
(未使用の IP アドレスすべて)
詳細設定(D)...
TCP ポート(T):
80
SSL ポート(L):
接続
接続のタイムアウト(N):
120
秒
HTTP Keep-Alive を有効にする(K)
ログの記録を有効にする(E)
アクティブ ログ形式(V):
W3C 拡張ログ ファイル形式
プロパティ(P)...
OK
キャンセル
適用(A)
ヘルプ

9.6 SOAP エラーの対処方法

エラー報告の 1 行目は、SOAP の仕様によって管理されます。無効な要求や呼び出し元の意図が判定できない要求に対して、SOAP のフォールト報告とフォールトコードを返します。SOAP からのエラーメッセージは一般的に *Fault* 要素内に含まれています。*Fault* 要素が存在する場合、*Body* 要素の子要素として表示される必要があります。*Fault* 要素が SOAP メッセージ内で表示されるのは一度のみです。SOAP *Fault* 要素には次のサブ要素が含まれます。

サブ要素	説明
<faultcode>	フォールトを識別するコードを表示します。
<faultstring>	人間が読み取ることができるフォールト説明を表示します。
<faultactor>	誰がフォールトを発生させたのかに関する情報を表示します。
<detail>	*Body* 要素と関連するアプリケーション固有のエラー情報を保持します。

faultcode 要素内でフォールトを説明するために使用する定義されたフォールトコード値は次のとおりです。

エラー	説明
VersionMismatch	SOAP *Envelope* の要素に対して無効なネームスペースを見つけました。
MustUnderstand	「1」に設定された *MustUnderstand* 属性をもつ *Header* 要素の直近の子要素は理解されませんでした。
Client	メッセージが間違って形成された、または間違った情報を含んでいました。
Server	サーバで問題があったため、メッセージを続行できませんでした。

9.7 SOAP デバック

9.7.1 Job API クライアント

Job API クライアント側のデバック用にメッセージロギングを有効にするためには、*stdsoap2.h* 内にある *#define DEBUG* 命令のコメントを削除し、クライアントアプリケーションをコンパイルします（または単に、*g++ -DDEBUG ...* を使ってデバックを有効にした状態でコンパイルします）。クライアントアプリケーションを実行すると、次の 3 つのファイルにそれぞれの動作を記録します。

– *SENT.log*
  アプリケーションによって送信された SOAP の内容です。

– *RECV.log*
  アプリケーションによって受信された SOAP の内容です。

– *RECV.log*
  アプリケーションにより実行された様々な動作を含むログです。

ログ記録のため、Job API アプリケーションの動作が遅くなる場合があります。

9.7.2 Job API サーバ

追加のデバック情報を入手するため、Job API サーバ側でもメッセージログ記録を有効にできます。詳しくは、*ReadMe.txt* ファイルをご覧ください。

ログ記録のため、Job API アプリケーションの動作が遅くなる場合があります。

10　Job API クライアントの開発

この章では、Job API サービスと相互運用できる gSOAP Web サービス開発環境を使った Job API クライアントアプリケーションの開発手順について説明します。

原則として、所定の WSDL に基づくクライアントアプリケーション用インタフェースを生成する様々な SOAP 開発ツールがあります。SOAP ツールに応じた選択は、使用するプラットフォームやプログラム言語（例：C、C++、Java など）に影響されます。gSOAP は、C と C++ を使用して、効率的な SOAP/XML Web サービスと Web サービスクライアントアプリケーションをデプロイメントするための開発環境で、プラットフォームには依存しません。したがって、原則として SOAP プロトコルを詳細に理解する必要はありません。

最初に、「gSOAP toolkit」を次からダウンロードしてインストールします。

http://www.cs.fsu.edu/~engelen/soap.html

（Job API Web サービスの開発には、リリース V2.7.13 の gSOAP が使用されました。）

gSOAP を使ったクライアントアプリケーションの構築を始めるには、次が必要です。

– C または C++ コンパイラ

– gSOAP soapcpp2（例：Windows 用 *soapcpp2.exe*）スタブおよびスケルトンコンパイラ

– *soapcpp2* スタブとスケルトンコンパイラに対して WSDL を gSOAP 仕様ヘッダファイルへ変換する gSOAP wsdl2h（例：Windows 用 *wsdl2h.exe*）WSDL パーサ（構文解析ツール）

– Web サービスアプリケーションとリンクするためのランタイムライブラリを持つ次のファイル：*stdsoap2.c* または *stdsoap2.cpp*、*stdsoap2.h*

– SSL（HTTPS）と圧縮処理ができるように、オプションとして Open SSL と Zlib ライブラリをインストールする必要がある場合もあります。

10.1 C/C++ スタブの生成

一般的に、SOAP クライアントアプリケーションの実行には、クライアントアプリケーションを起動するために必要な各リモート方式用のスタブルーチンが必要です。スタブの主な役割とは、パラメータデータを marshal し、パラメータを使って指定された SOAP サービスへ要求を送信して応答を待ち、応答が来たら応答のパラメータデータを unmarshal することです。クライアントアプリケーションは、ローカル方式を起動するかのように、リモート方式に対するスタブルーチンを起動します。gSOAP *wsdl2h WSDL* パーサ、*soapcpp2* スタブ、およびスケルトンコンパイラは、Web サービスクライアントの開発を自動化します。gSOAP スタブとスケルトンコンパイラは、SOAP クライアントを構築するために必要なソースを生成するプリプロセッサです。*soapcpp2* への入力は、標準 C/C++ ヘッダファイルにより構成されています。このヘッダファイルは、あらかじめ gSOAP WSDL を使ってサービスの WSDL 文書から作成する必要があります。

Job API クライアントを開発するためには、最初に Job API の WSDL を gSOAP WSDL を使ってインポートします。SOAP リモート方式の Job API 関数プロトタイプの C/C++ 宣言を使ってヘッダファイルを作成します。

```
wsdl2h -o JobAPI.h JobAPI.wsdl
```

これで Job API WSDL から C++ 形式でファイル *JobAPI.h* が生成されます。ヘッダファイルを生成し純粋な C クライアントアプリケーションを開発するには、次のコマンドを実行します。

```
wsdl2h -c -o JobAPI.h JobAPI.wsdl
```

次の開発手順では、gSOAP コンパイラは、クライアントアプリケーションに対するスタブルーチンへこの生成された *JobAPI.h* ヘッダファイルを変換します。結果として生じるスタブルーチンは、C および C++ クライアントアプリケーションが既存の Job API Web サービスとスムーズにやり取りできるようにします。gSOAP コンパイラは次のコマンドラインから起動します。

```
soapcpp2 JobAPI.h
```

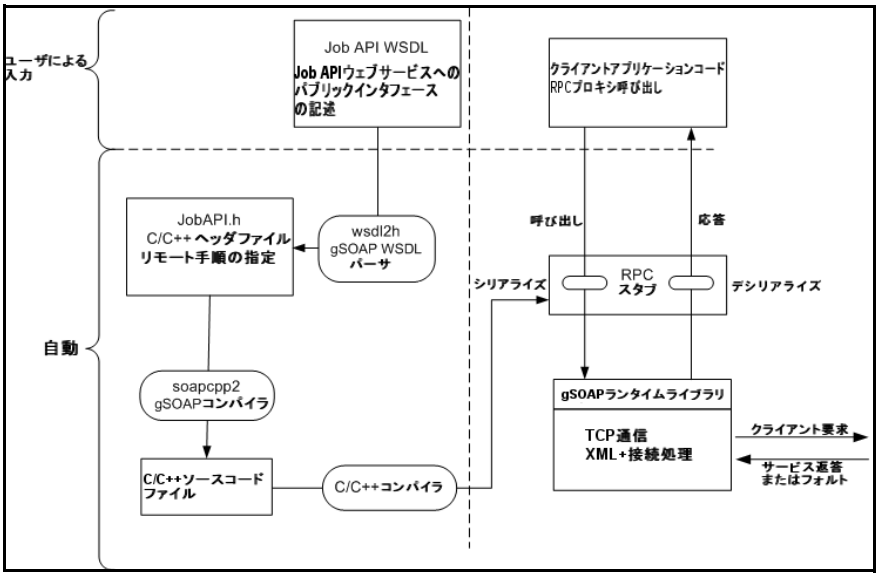
この Job API SOAP サービス方式の入出力パラメータは、単純なデータタイプまたは複合的なデータタイプの場合があります。これらは、WSDL に含まれているデータタイプの定義に応じて WSDL パーサによって生成されます。さらに、gSOAP スタブとスケルトンコンパイラは自動的にデータタイプのシリアライザとデシリアライザを生成します。そして生成されたスタブルーチンが XML でリモート方式のパラメータ内容を暗号化および複合化できるようにします。

WSDL パーサまたはスタブ / スケルトンコンパイラにより生成されたファイルの詳細は次のとおりです。

soapStub.h	入力ヘッダファイルから作成された修正または注釈付きファイルです。
soapH.h	すべてのクライアントとサービスソースに含まれるメインヘッダファイルです。
soapC.cpp	指定したデータ構造のシリアライザとデシリアライザです。
soapClient.cpp	リモート操作用クライアントスタブルーチンです。
soapServer.cpp	サービススケルトンルーチンです。
soapClientLib.cpp	ローカルの静的（デ）シリアライザと結合したクライアントスタブです。
soapServerLib.cpp	ローカルの静的（デ）シリアライザと結合したサービススケルトンです。
JobAPI.h	SOAP リモート方式の Job API 関数プロトタイプの C/C++ 宣言を含むヘッダファイルです。
JobAPISoapBinding.nsmp	サーバとクライアントが収容するネームスペースマッピングテーブルです。
soapJobAPISoapBindingProxy.h	C++ プロキシオブジェクトです（*soapClient.cpp* の代わりとして使用できます）。
soapJobAPISoapBindingProxy.cpp	C++ プロキシオブジェクトの実行です。
soapJobAPISoapBindingObject.h	C++ サーバオブジェクトです（*soapServer.cpp* の代わりとして使用できます）。
soapJobAPISoapBindingObject.cpp	C++ サーバオブジェクトの実行です。
.xsd	コンパイラに対するヘッダファイル入力のデータ構造により使用される各ネームスペースの接頭辞 ns 用の XML スキーマを使って、*ns.xsd* ファイルは生成されます。

表 2:

Job API クライアントアプリケーションの開発プロセスに関する全手順は次の図のとおりです。



10.2 生成された C/C++ スタブの使用

生成された C/C++ スタブルーチンを使用するには、次のような C/C++ クライアントアプリケーションコードを実行します。

```
// contens of clientJobAPI.cpp

#include "soapJobAPISoapBindingProxy.h"
#include "JobAPISoapBinding.nsmap"
#include <stdio.h>
#include <string.h>


int main()
{
    int retcode(0);
    // URL of Job API on local host with IIS
    string sEndPoint("http://localhost/scripts/JobAPI/JobAPI.exe");
    // URL of Job API on local host with ServerView installed Apache Web server
    // string sEndPoint("http://localhost:3169/scripts/JobAPI/JobAPI.exe");

    JobAPISoapBinding jobAPI;
    soap_init(jobAPI.soap);
    jobAPI.endpoint = (char*)sEndPoint.c_str();

    // Job API remote methode GetAPIVersion
    struct ns2__GetAPIVersionResponse getAPIVersionResponse;
    char* userKey = "myUserKey";

    jobAPI.ns2__GetAPIVersion(NULL, getAPIVersionResponse);
    if (jobAPI.soap->error != SOAP_OK)
            soap_print_fault(jobAPI.soap, stderr);                                  \
    else
            if (getAPIVersionResponse._ErrorCode != 0)
            {                                                                       \
                    std::cout << "ErrorCode: 0x" << hex << getAPIVersionResponse._ErrorCode
                            << ", ErrorString: " << getAPIVersionResponse._ErrorString << endl;
            }
            else
            {
                    std::cout.setf(ios_base::hex, ios_base::basefield);
                    std::cout << "APIVersion: " << getAPIVersionResponse._APIVersion << endl;
            }

    soap_end(jobAPI.soap);
    soap_done(jobAPI.soap);
    return 0;
}
```

C++ コンパイラを使ったコンパイル後、例えば g++ の場合：

```
g++ soapC.cpp soapClient.cpp stdsoap2.cpp client.cc -o client
```

アプリケーションを実行すると、Job API メソッド GetAPIVersion を呼び出し、アクセスしたホスト（ローカルホスト以外）にインストールされている Job API サービスのバージョンを返します。

このクライアント開発に関する説明では、詳細が省略されています。詳しくは gSOAP ユーザーズガイドをご覧ください。

索引

A
AddGroupMember 108
AddRepository 74
AddServer 33
API エラーコード 149
API 管理 28

C
CancelJob 137
ChangeServerConfiguration 40
CreateDirectory 93
CreateGroup 100
CreateImage 79
CreateUserKey 32

D
DeleteDirectory 95
DeleteGroup 102
DeleteImage 87
DeleteScheduledJob 143
Deploy 118

E
ErrorCode 149
ErrorString 149
ExportDeploymentConfiguration 57

G
GetAPIinfo 29
GetAPIVersion 28
GetBootMode 70
GetDirectoryContents 97
GetGroupList 103
GetGroupMemberList 111
GetGroupMemberProperties 115
GetGroupProperties 107
GetImageProperties 92
GetJobStatusDetails 131
GetPower 68
GetRepositoryContents 88
GetRepositoryList 77
GetScheduledJobLogList 141
GetScheduledJobProperties 139
GetServerJobStatus 134
GetSystemInfoResult 48
Group Management 99

I
ImportDeploymentConfiguration 55
Installation process 152

R
RdDeplServerList 49
ReadImagefilenameList 90
ReadImageInfoText 91
RemoveGroupMember 110
RemoveRepository 76
RemoveUserKey 60
Repository Management 72

S
SetGroupMemberProperties 113
SetGroupProperties 105
SetPower 61
SetPXEBootImage 65
SetUserKey 59
StartSystemInfoRetrieval 45

い
インストール 151

か
関数
- AddGroupMember 108
- AddRepository 74
- AddServer 33
- CancelJob 137
- ChangeServerConfiguration 40
- CreateDirectory 93
- CreateGroup 100
- CreateImage 79
- CreateUserKey 32

DeleteDirectory 95
DeleteImage 87
DeleteGroup 102
DeleteScheduledJob 143
Deploy 118
ExportDeploymentConfiguration 57
GetAPIinfo 29
GetAPIVersion 28
GetBootMode 70
GetDirectoryContents 97
GetGroupList 103
GetGroupMemberList 111
GetGroupMemberProperties 115
GetGroupProperties 107
GetImageProperties 92
GetJobStatusDetails 131
GetPower 68
GetRepositoryContents 88
GetScheduledJobProperties 139
GetRepositoryList 77
GetScheduledJobLogList 141
GetServerJobStatus 134
GetSystemInfoResult 48
ImportDeploymentConfiguration 55
RdDeplServerList 49
ReadImagefilenameList 90
ReadImageInfoText 91
RemoveGroupMember 110
RemoveRepository 76
RemoveUserKey 60
SetGroupMemberProperties 113
SetGroupProperties 105
SetPower 61
SetPXEBootImage 65
SetUserKey 59
StartSystemInfoRetrieval 45
関数の概要 21

さ
サーバ / ノードリスト管理 33

し
システム要件 151
ジョブ管理 130
ジョブスケジューリング 145

て
ディレクトリ管理 93
データの種類 26
デプロイメント管理 117
電源管理 61